THE USE OF SIMULATION TO INVESTIGATE HEURISTICS FOR ALTERNATE ROUTING IN MANUFACTURING SYSTEMS

Thesis submitted by

Susan Eileen Grinsted

to the University of Warwick for the award of Ph.D by research for work carried out in the Engineering Department at the University of Warwick

September 1990

TABLE OF CONTENTS

Appendices

LIST OF TABLES

LIST OF FIGURES

Acknowledgements

Many friends and colleagues have supported me in this work but particular thanks are due to my supervisor, John Hill, for his continued support, to Rajat Roy for his guidance on simulation practice, the members of the Simulation Group, Prof.S.K.Bhattacharyya for providing equipment at a time of need, and finally to my husband S.G. for his enduring patience and sanity without which the work would not have been completed.

Declaration

I declare that I am the sole author of this work and that it has not been published before in any form or used in any other submission for any other award.

The simulation model structure and associated program code used for experimentation are based on a model written by Rajat Roy, Simulation Manager, Manufacturing Systems Engineering Group, Warwick University, and myself. The model has been substantially modified by me for this work. It would have been very difficult to revise such a complex model if I had not taken part in all its stages of development.

Summary

Most approaches to classical scheduling problems are unable to cope with alternate routes, but simulation is one approach that may be used. Similarly, most computer aided production management systems are unable to handle alternate routes because of the increased complexity in file structure and management. On the shop floor however, alternate routes are used to meet short-term capacity problems, although some managers feel that quality may be compromised and complexity increased.

Increased awareness of the benefits of flexibility must now question routing flexibility and subsequently flexibility in operation sequence.

Characteristics of alternate routing are stated, and rules for choosing an alternate route for a job are formulated. A simulation model was developed of a simple machine shop comprising up to six machines to process batches of four component types. Setting up times, breakdown patterns and inter-operational transport time are included in the model.

Using the simulation model, a number of rules were tested on single operation alternates and partial route alternates. Two rules work well for different purposes. On single operation alternates, a forced ratio rule achieves workload balance while a rule which examines waiting workload achieves the shortest flowtimes. It is shown that balancing workload on partial routes can incur a penalty on flowtime. Good flowtimes on partial routes may be achieved by using a forced ratio rule or by examining waiting workloads.

1. Introduction to Alternate Routes

The provision of alternate routes of manufacture in a traditional batch-operated machine shop is widely believed to improve the level of due date achievement and reduce throughput time. The rules which should be used to divert jobs away from the preferred route onto an alternative route, and the rules governing the choice of one route from several options for a particular batch under a particular set of circumstances are not widely known.

1.1 Importance of alternate routing capability

The need to establish a workable means of handling alternate routes is now being emphasised by computerised manufacturing systems, especially in batch manufacturing. Computer controlled machining stations are being linked together by automatic material handling systems. These machining stations are becoming increasingly capable and adaptable. An opportunity is available to increase facility flexibility by scheduling workstations in real-time, reflecting the current state of the whole system. The number of possible alternate routes has therefore increased.

The ability to carry out an operation on a machine other than the preferred machine offers a number of advantages in practice:

1.1.1 The disturbance to flow of work caused by delays such as long set up times or breakdowns can be lessened, by diverting some jobs to other facilities.

1.1.2 The number of set up changes may be reduced by designating different alternative routes or machines as preferred routes for different jobs.

1.1.3 Provision of alternates provides extra capacity to relieve a bottleneck operation or machine, reducing waiting time at the bottleneck, thus reducing flow time and reducing possible subsequent shortages.

1.1.4 The "make or buy" decision can be viewed as a special case of alternate routes (1) where for example, "make" may be the preferred route, for comparison with the "buy" route. This case will not be examined.

This work has been undertaken to establish the best workable heuristics for a stated set of operating conditions and subject to certain practical constraints.

1.2 Definition of Alternate Route

According to a glossary of terms for computer integrated manufacturing (2) the term "alternate routing" is used to describe "an alternate method or sequence of performing an operation, a series of operations, or a complete routing. The alternate is generally used because of a machine breakdown or an excessive overload on the machines or work centres". The term "alternate route" has been used to describe two distinct cases of flexibility in scheduling, which are demonstrated in the following section by early work in this area.

It should be noted that "alternate" is the North American term used where "alternative" is strictly correct in English. "Alternate" is usually taken in British English as meaning to do things by turns, whereas "alternative" is used to mean that options are mutually exclusive. However, common usage decreasingly distinguishes these forms, and both terms will be used interchangeably throughout the thesis.

1.2.1 Russo's definition

Russo (3) considers a manufacturing route to comprise a number of partial routes, each of one or more operations (fig.1). Each operation within the partial route must be completed before the next partial route may be started but the order of operations within the partial route is not significant.

In fig.1, operation 1 must be performed first. Operations 2 and 3 can be performed in any order, but neither can be started before operation 1 has finished, and both must be completed before any subsequent operations may be started. Operation 1 has no alternates. Operations 2 and 3 each have one alternate. In this example, operations k-1, k and k+1 each have two alternates.

1.2.2 Wayson's definition

In job shop scheduling theory, the job is considered to be a list of operations o_{ij}. Each operation is an ordered pair where i is the operation number on this job and j is the machine. Much scheduling theory depends on a fixed relationship between operation and machine. Wayson's (4) definition for "alternate route" lifts this restriction by defining a list of alternate machines for each machine in the shop. On completion of an operation, the scheduler has the option of assigning the job to the specified machine for the next operation or any of the machines on the list of alternates. For example, schedule A in figure 2 represents fixed routing (i.e. no alternates) and schedule B indicates that a proportion of jobs may be sent to an alternate.

The row index represents the specified machine, and the column index represents the alternate machine. The entry p_{ij} indicates the availability of the alternate at

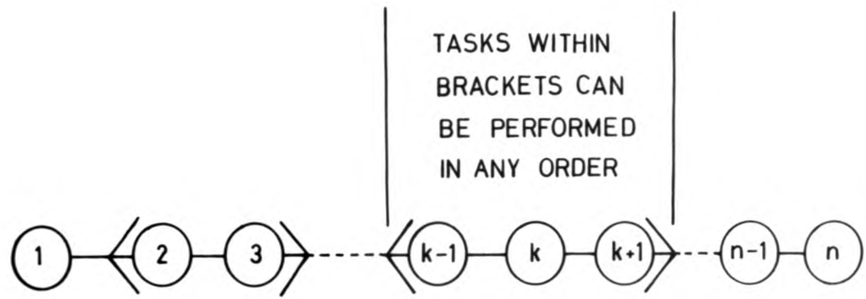


FIG. 1 PARTIALLY ORDERED TASKS (RUSSO)

Figure 2 Alternate machine availability matrices (Wayson, 4)

Schedule A

		Alternate machine no.						
		1	2	3	4	5	6	7
	1	1	0	0	0	0	0	0
Preferred	2	0	1	0	0	0	0	0
Machine	3	0	0	1	0	0	0	0
	4	0	0	0	1	0	0	0
	5	0	0	0	0	1	0	0
	6	0	0	0	0	0	1	0
	7	0	0	0	0	0	0	1

Schedule B

		Alternate machine no.						
		1	2	3	4	5	6	7
	1	1	1	½	0	0	0	0
Preferred	2	0	1	1	½	0	0	0
Machine	3	0	0	1	1	½	0	0
	4	0	0	0	1	1	½	0
	5	0	0	0	0	1	1	½
	6	½	0	0	0	0	1	1
	7	1	½	0	0	0	0	1

each decision point. For example, under schedule B, the alternates to machine 3 in schedule B are (i) always machine 4 and, (ii) machine 5 for 50% of the decisions.

1.2.3 Unisymmetric and bisymmetric matrices

Wayson also distinguishes between bisymmetric and unisymmetric matrices. A bisymmetric matrix implies that if machine A is an alternate to B, then B is naturally an alternate to A. This relationship does not necessarily hold for unisymmetric matrices, such as schedule B. In practice for example, a group of nominally identical machine tools may form a bisymmetric matrix. However presses, where a job may be run on a more powerful press but not on a less powerful press, would be part of a unisymmetric matrix. This work considers both these types of relationships.

1.3 Characteristics of alternate route decisions

Characteristics of both definitions are that at certain points during the course of operations on a job, the next "operation number-machine number" pair has to be chosen from a range of options. Thus, of the different inter-operational path types that could occur in a network representing the flow of a job through a shop (one to one, one to many, many to many, many to one), this is a one-to-many or part of a many-to-many type decision. Under certain circumstances therefore, the likelihood of other jobs considering one or more of the same alternates as a job about to make a decision now may need to be taken into account.

1.4 General cases

In this work, the second definition of alternate routes will be assumed where a job may be directed to one of a number of alternative machines for the next operation. It is considered that this is directly analogous to the first definition, where the next operation is chosen for a job from a number of alternative operations, if those operations could be carried out on different machines, according to the workload on those machines.

A number of different cases in alternative routing can be identified.

1.4.1. Case A: New technology

In case A (see fig.3), a new facility Op 20 has replaced Ops 30, 40 and 50 but old facilities remain to take peak loads. All jobs of this type require to undergo Op 10, and Op 60, but may be processed through either Op 20, or Ops 30, 40 and 50. Although it may be preferable to route jobs through Op 20, under certain circumstances (e.g. breakdown, tool breakage, build up of work waiting, several high priority jobs etc.) it may be advisable or necessary to choose the alternative route.

This case illustrates a number of points. There is a common starting point, Op 10, and a common finish point, Op 60. The partial routes are therefore contained or confined by the start and finish points which may be an operation or a many to many node. There could be one shared storage area holding work available for both routes if the machines are located close to one another or there could be separate storage areas if the machines on different routes are dispersed.

FIG. 3 CASE A – NEW TECHNOLOGY

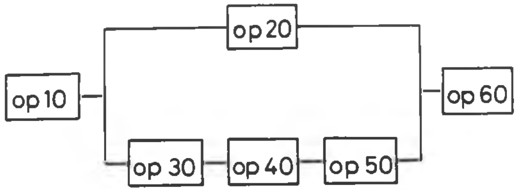
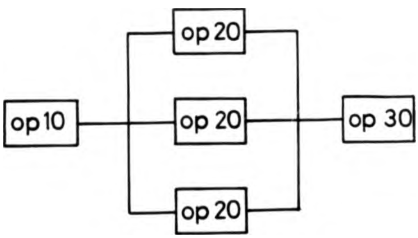


FIG. 4 CASE B – 'SINGLE OP' OPTIONS



1.4.2 Case B: "Single op" options

In case B (see fig.4), Op 20 may be carried out on machine 1 or machine 2 or machine 3. These machines may or may not be identical (but all are dedicated to Op 20). It may be that one machine is considered to be more 'efficient' than the others, and hence may be preferred.

This case illustrates the need to recognise identical or non-identical machines, and possible preferred routes. Similar to Case A, storage areas for arriving work could be pooled or dispersed. The partial routes are also confined with a common start and finish point.

1.4.3 Case C: Mixed Capabilities

In case C (see fig.5), non-identical machines 1 and 2 may be capable of different operations

e.g. machine 1 - Ops 10 and 20
machine 2 - Op 20
machine 3 - Op 30

To distinguish the conditions from those of multiple facilities, the principle characteristic is a choice of path from operations that are non-identical. Hence one option must be better than another. It may be argued that even 'identical machines' are not identical in practice, for example due to different reliability histories.

FIG. 5 CASE C – MIXED CAPABILITIES

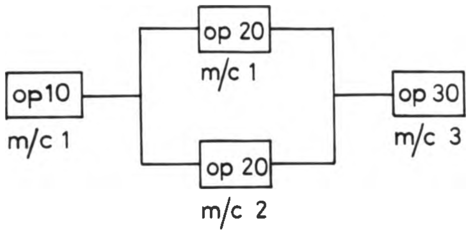
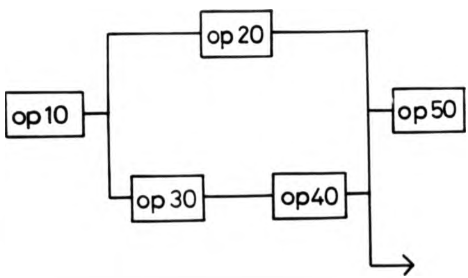


FIG. 6 CASE D – SHARED/UNCONFINED BRANCHES



1.4.4 Case D: Shared/unconfined branches

In case D (see fig.6), the partial routes are shared but do not have a common start and finish point. In this case it is unlikely that the operations on the "alternative partial routes" are really alternatives if subsequent operations are determined by which partial route is selected. This case will not be considered.

1.5 Significance of the alternate route decision

This "where next?" decision is one of two local scheduling decisions which govern the flow of work through a collection of workstations. The other decision, "what next?", which chooses the next job for a machine from the queue of waiting jobs by means of dispatching rules, (also called sequencing or priority rules), has been extensively studied. Under conditions of negligible inter-operational transportation time and unlimited in-process storage, the "where next?" decision is not needed even if alternative routing is allowed. (This is described in section 1.6.3 as a level 3 decision.)

1.6 Timing of the "where next?" decision

Russo (3) identifies 3 levels at which the choice between alternates may be decided:

1.6.1 Level 1

The choice between any alternates is made before the job is launched into the shop. The net effect is to produce fixed routing by preventing local conditions altering the course of the job once it is in progress. For a job of more than one operation, all flexibility in response to local difficulties can be assumed to be lost.

1.6.2 Level 2

A decision between alternative partial routes can be made at the time of transition between operations. As soon as an operation is completed, the state of each alternate route can be assessed and a choice made between the preferred route and an alternative. This technique can be thought of as "choose a queue according to some criterion", e.g. least work, and is thus a queue assignment problem. This type of decision is likely to be made in practice, based on a spot judgement by a local supervisor, taking into account factors such as the amount of work waiting, a preferred route, crewing restrictions and sequential set up dependencies.

1.6.3 Level 3

The highest level, and most efficient (3) decision is made immediately prior to starting the next operation. In modelling terms, the job is placed in the queue of jobs at every machine where the operation could be carried out, allowed by an assumption that inter-operational transport times are negligible and in-process storage is unlimited. Under these circumstances, the alternate routing problem does not exist. The outcome will depend on the dispatching rules being used. In practice of course, this assumption is only valid where all the first operations draw work from the same local buffer stock, with approximately the same ease. If the locations of the first operations are distributed, and transport times are more than negligible, this highest level decision becomes unworkable.

Because of the conditions commonly prevailing in practice, this work concentrates on level 2 type decisions.

1.7 Summary of conditions

The main factors and characteristics of this problem are:

1.7.1 Alternate routing is considered to comprise alternative partial routes within the list of operations such that selection of one partial route, p', will remove the need to carry out any of the operations on any of the alternative partial routes.

1.7.2 Alternate routes may be unisymmetric or bisymmetric i.e. if A is an alternative to B, the bisymmetric case states that B must be an alternative to A. The unisymmetric case does not require this.

1.7.3 Alternate partial routes must be constrained by a common start point and a common finish point. Between these points, the partial route may comprise one or more operations.

1.7.4 The "where next?" decision could be one-to-many or many-to-many.

1.7.5 Inter-operational transportation times may not be negligible.

1.8 Objectives of this work

The primary objective of this work is to provide a method or methods which will enable a good decision to be made between alternate routes, at the point of transition between operations, subject to constraints commonly prevailing in practice. Shop floor work reporting, tracking and monitoring systems allow routing decisions to take the state of the manufacturing network into account.

The propositions to be tested are therefore:

1.8.1 That it is feasible to define decision rules to enable the next machine to be chosen from available alternatives using data which should be available from a shop floor work reporting system.

1.8.2 That different rules will be more effective against different performance criteria.

1.8.3 That different rules may be more appropriate to the different cases stated in 1.4.1-1.4.4.

1.8.4 That different rules may be more appropriate to preferred routes in contrast to alternative routes which are equally weighted.

2. Review of the development of scheduling theory with particular regard to the treatment of alternate routes

2.1 Introduction

Scheduling 'problems' have been studied by mathematicians and operational research scientists for over 30 years. In addition to the massive literature published on scheduling, it may be also be assumed that an enormous quantity of unpublished work has been carried out by those involved in seeking workable solutions to particular industrial conditions.

It is proposed here that there have been few major breakthroughs in the mathematical approach to scheduling in the last 30 years which could be adopted by a manual or computer scheduling system for practical use. The major developments and approaches will be summarised in this chapter. As a mathematical problem, the quality of many of the proposed solutions is not to be belittled and one of the most important advances in theoretical development was the understanding of the real complexity of the problem, and hence the requisite complexity of any solution. However, as an aid to the industrial scheduler, it must be argued that the objectives of the mathematician and the industrial scheduler are at variance. Many industrial schedulers still adhere to the concept of optimality of a schedule. In most industrial situations however, conditions change too fast and the number of variables so far exceeds the number of points of control, that the goal should be 'good approximate solutions' which will yield a reasonable chance of obtaining a good schedule and minimise the chance of a poor solution.

In this situation, the visibility and comprehensibility of heuristics are attractive.

In parallel with the disappointing realism of scheduling theory, it must be stated that scheduling is still a neglected area of industrial management and hence a much under-used source of productivity improvement. It may be partly as a result of the well-known complexity of the problem, or it may be one of the less approachable areas of management in comparison to method changes, design changes, bonus schemes or quality improvements for example.

2.2 Description of the 'scheduling problem'

The overall task of scheduling is to commit resources (machines, "facilities") to jobs (tasks, orders) such that the performance objectives are met in the best way possible under the circumstances.

Much scheduling theory has developed from the job-shop case, which may be considered to be an extreme example of planning a series of jobs to be processed through a group of machines. In general, each job requires a number of operations in a pre-defined order on a number of machines. The objective is to obtain a feasible schedule of operations that will minimise a 'regular' measure of performance (i.e. the measure of performance only increases if one of the job completion times increases in comparison with another schedule).

2.3 General constraints and assumptions

Much early rationalisation of the job shop problem is attributed to Gere (5) who brought together and examined a number of basic heuristics and definitions from work carried out in the fifties. Mellor (6) summarises the typical simplifying assumptions made by job-shop theorists which were listed by Gere (5), who himself based the list on Sisson (7,8).

2.3.1 No machine may process more than one operation at a time.

2.3.2 Each operation, once started, must be performed to completion (no pre-emptive priorities).

2.3.3 Each job, once started, must be performed to completion (no order cancellations).

2.3.4 Each job is an entity; that is, even though the job represents a lot of individual parts, no lot may be processed by more than one machine at a time. This condition rules out assembly operations.

2.3.5 A known, finite time is required to perform each operation and each operation must be completed before any other operation which it must precede can begin (no "lap-phasing"). The given operation time includes set-up time.

2.3.6 The time intervals for processing are independent of the order in which the operations are performed. (In particular, set-up times are sequence-independent and transportation time between machines is negligible).

2.3.7 In-process inventory is allowable.

2.3.8 Machines never break down and manpower of uniform ability is always available.

2.3.9 Deadlines (due-dates), if they exist, are fixed.

2.3.10 The job routing is given and no alternative routings are permitted.

2.3.11 There is only one of each type of machine (no machine groups).

2.3.12 All jobs are known and are ready to start processing before the period under consideration begins. (This is the "static" scheduling problem).

Many of these assumptions appear to be unreasonable at first sight, and have been listed here in order to state the basic case. Many researchers have relaxed one or more of these assumptions in order to examine variants of the job shop problem. To study alternate routing, rule 10 must be ignored and, in some cases, rule 11 also. Rules 6 and 7 are significant because transportation times and limited inventory are integral to the validity of alternative routing and current production control objectives respectively.

The pure job-shop case has been useful for mathematically proving theorems governing optimality in the general case, and for controlled relaxation of one or more constraints.

2.4 Nature of the solution to scheduling problems

Under the assumptions in section 2.3, the solution to the job-shop problem is the sequence in which the known jobs should be processed. Scheduling and sequencing are thus synonymous, mainly due to the negligible inter-operation transport times, unlimited in-process inventory, and restrictions on alternative routes.

However, although much scheduling investigative work is subject to constraints which are too simplistic or rigorous for practical use, some useful general guidelines for preparing schedules have been produced and will be discussed shortly.

2.5 Measures of performance

Having described the basic problem, performance objectives are required against which success can be measured. Mellor (6) stated that most sequencing systems could be put into one of two classes - either those whose broad objective is to minimise "makespan" (i.e. the total time required to process a pre-defined set of jobs) or those whose objective is to minimise lateness (i.e. achieve due dates).

Typical criteria cited by Mellor (6) from Gere (5) for the first class are:

2.5.1 Finish the last job as soon as possible; i.e. minimise the interval of time from start of processing until all jobs are completed.

2.5.2 Finish each job as soon as possible. (Minimise the sum of completion times).

2.5.3 Minimise the in-process inventory.

2.5.4 Maximise machine utilisation.

For the second class:

2.5.6 Minimise the number of late jobs.

2.5.7 Minimise the total tardiness.

2.5.8 Minimise the costs due to not meeting due dates exactly. This includes the first two as particular forms of the cost function. Costs due to earliness might be included.

Mellor did acknowledge a third set of performance criteria - those based on costs. Blackstone et al (9) emphasise that the only relevant measure of performance for a dispatching rule is cost effectiveness. Dispatching rules influence delay costs, inventory costs and set up costs, but particularly delay costs. However, in practice the structure of delay costs varies widely between companies and non-cost performance measures are used such as flow time and tardiness which could feasibly be extrapolated to a cost using a suitable function. Mellor listed 27 measures of performance considered by Beenhakker (10) who assigned coefficients to each performance measure in order to construct an appropriate pay-off function using utility theory.

Mellor's list includes the following additional measures:

2.5.9 Minimise facility set-up costs.

2.5.10 Day-to-day stability of work force.

2.5.11 Adherence to promised shipping dates.

2.5.12 Maximum output (production rate)

2.5.13 Minimum materials-handling cost.

2.5.14 Adherence to arbitrary job priorities, such as arise in dealing with preferred customers, emergency repair parts etc.

2.5.15 Technological feasibility. (i.e. operation sequence)

2.5.16 Sensitivity to possible production changes.

2.5.17 Reserve capacity for rush orders.

2.5.18 Maximum utilisation of manpower.

2.5.19 Optimal assignment of various labour grades.

2.5.20 Minimum raw material inventory.

2.5.21 Minimum finished product inventory.

2.5.22 Minimum obsolescence and deterioration of products.

2.5.23 Shortest make-span for certain products.

It is significant that despite this comprehensive and reasonable list of performance criteria, the success of any one approach to a scheduling problem is almost exclusively measured against one performance objective only. In practical circumstances, this is rarely the case and a balance must be found between conflicting objectives, where one objective can often be improved at the expense of another.

One valid observation made by Conway, Maxwell and Miller (11), is that "in some models, it has been possible to find optimal procedures only by departing from what would be considered to be the most natural and realistic criteria". Despite their extensive treatment of different measures of performance they observe that the "ultimate" measure of performance is total job waiting time since any waiting time is undesirable and will adversely affect most other measures.

2.6 Approaches to the scheduling problem

Despite the known restrictions on alternate routing in job shop scheduling problems, it is useful to summarise the varied approaches that have been made over the last 30 years. A classification of problem characteristics and also problem approaches will be given.

2.6.1 Classification of problem characteristics

There are many classifications of scheduling problems but a useful classification of problem characteristics is given by Graves (12), which has been used to generate fig.7.

Five dimensions to the problem are listed. The list of required jobs can be a list of external customer orders (i.e. make to order and there are no sales from stock), called an open shop, or a closed shop where internal orders (jobs) are raised to replace sales from inventory (and hence reorder points, reorder quantities and other reordering mechanisms become critical). The closed shop problem depends on lot-sizing of the requirements for processing through different facilities and will not be considered here. The processing complexity of the models used to investigate scheduling problems have been divided into one-stage and multi-stage i.e. single or multiple operations required per job. The scheduling criteria dimension can be split broadly into cost and performance measures. It can be seen that the specification (i.e. processing time in simple models) can either be deterministic or stochastic, and the scheduling environment may be static or dynamic. As Graves points out, it is interesting to note that most models for scheduling problems are deterministic and static, while most practical production environments are stochastic (tool breakage, machine breakdown, operator performance, etc) and dynamic (jobs continually arriving).

2.6.2 Classification of problem approaches

A classification of approaches to the scheduling problem is shown in table 1, starting from the classification of approaches given by Conway, Maxwell and Miller (11) into algebraic, probabilistic, and Monte Carlo methods. Most

FIG. 7 CLASSIFICATION OF CHARACTERISTICS OF SCHEDULING PROBLEMS (GRAVES)

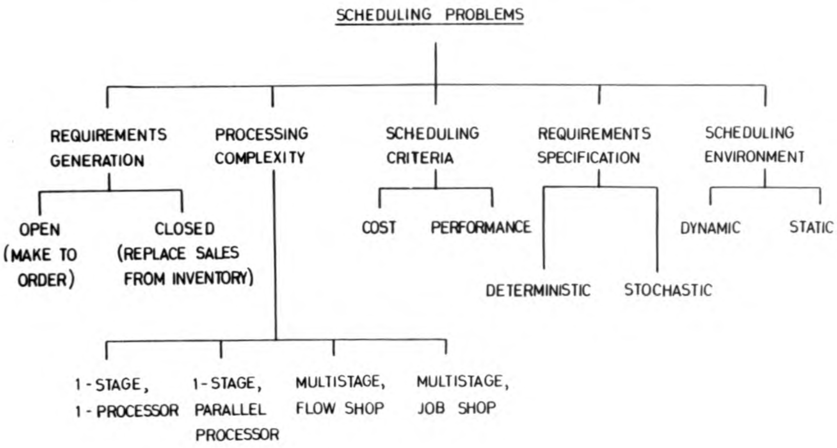


Table 1 Classification of Approaches to the Scheduling Problem

Nature of solution	Broad approach	Techniques	
Non-optimal solutions	Monte Carlo	Priority rule	
		Heuristics	Simulation
Optimal solutions	Algebraic	Enumeration	Dynamic Programming
			Branch and bound
			Integer programming
			Complete Enumeration
		Constructive Algorithms	e.g. Johnson's rule
	Probabilistic	Queuing Networks	
		Critical path - PERT/CPM	

scheduling investigation has taken the algebraic branch and examines deterministic, static problems. The usefulness of some of these approaches to the alternate routing problem will be discussed in turn, starting with the single operation, single machine case.

2.7 Single operation, single machine problems

The single machine case has more practical value than might be assumed initially, because it could represent a shop where one machine acts as a "bottleneck", thus determining the flow of work through subsequent operations. A single machine problem may yield at least a first approximation to the solution.

A scheduling tactic in more complex situations is to "insert idle-time" i.e. deliberately keep a machine idle, to wait for an expected urgent job. However, it may be proved mathematically (11) that there is no need for idle time between any of the tasks in a static single machine problem. This is intuitively obvious since the machine is not dependent on any other machine for arrival of work, and clearly cannot be blocked by downstream operations, and all the jobs are known and present at the beginning of the period. Hence the task of the scheduling process is to find the best sequence of jobs, or permutation schedule, against the performance criteria.

Another schedule tactic used in more complex problems is pre-emption, where a job is removed from a machine before processing is completed in order to start a more urgent job. It can be shown that there is no advantage in using pre-emption in this case (11).

2.7.1 Static case

Simple ordering rules have been shown to yield optimal schedules under simple criteria. Smith (13) showed that average flow time would be minimised by ordering the jobs such that the job with the shortest processing time was processed first, and the longest was processed last. By equivalence, average completion times, average waiting time, average lateness, number of jobs still to be completed and number of jobs waiting will be minimised (see 11 for proof). It may be seen intuitively that this rule keeps work flowing through the shop. This SPT rule (shortest processing time, or SIO, shortest imminent operation) does not however address due dates.

Jackson (14) showed that maximum tardiness (amount of time by which due date is exceeded) would be minimised by processing the job with the earliest due date first, but this does not minimise average flow time.

Many hybrid rules and algorithms have been designed to effect a compromise between these conflicting objectives. The primary purpose of the research has been to find a priority rule, or means of ordering, which will optimise the schedule against a set of performance criteria. It must therefore be recognised that the choice of rule significantly depends on the performance criteria.

2.7.2 Dynamic case

A major benefit of the development of priority rules has been their ability to stand up to dynamic conditions. A very considerable body of literature exists on a wide range of priority rules and their relative performance under different conditions.

The most common and important rules are:

(i) SPT

(ii) Earliest due date

(iii) Least slack remaining (time remaining to due date less processing time)

(iv) Least slack per operation (slack time per remaining operation)

(v) Critical ratio (time remaining to due date divided by processing time remaining)

(vi) FIFO (first in, first out, also known as FCFS, first come, first served)

The performance criteria determine the success of any rule (9, 15, 16). In general, rules involving processing times (eg. SPT) address flow time and inventory criteria but perform badly against due date criteria. Rules involving due dates, including the slack based rules, improve due date achievement but compromise flow time. A rule which does not take into account either processing time or due dates, such as FIFO, performs in a similar manner to random selection, which performs neither as well as good processing time rules nor good due date rules with regard to the mean and variance of most criteria. FIFO is frequently used in research work as a 'control' against which the success of any other rule is measured.

Rule combinations have met with mixed success (15, 16, 17). It is interesting to note that critical ratio is most widely used in industry (12). In general, a combination of simple priority rules, or a combination of heuristics with a simple priority rule can work better than individual rules (9, 15, 16, 17). However, it has also been observed that results are influenced by characteristics of the shop under investigation, such as the level of machine utilisation and method of assigning due dates (9).

Many of these examinations are carried out in highly idealised models and it is questionable whether the improvement, for the increased effort, would be achieved in practice. It is noted that computing power has increased significantly since some of the work was carried out but as Gere (17) observed, if the simple rule is practically as good, then keep it simple.

2.8 Multi-stage, multiple machine problems

The flow time of a job through a simple shop comprises processing time and waiting time. The waiting time will depend on the priority given to this job in competition with other jobs through the facility, and on the provision of capacity to cope with the workload.

Conway et al (11) approached the multi-stage scheduling problem by means of two extreme cases:

(i) pure job shop - in which a job leaving a machine is equally likely to go to any other machine in the shop, and

(ii) pure flow shop - in which there is only one path through the shop that all jobs will follow.

To distinguish these cases from alternative routing, it should be noted that the whole route is decided before the first operation in both cases. Thus, there is no "choice" of route at the end of any operation.

A special case in multiple machine problems concerns the handling of parallel processors, or multiprocessors.

2.9 Multiprocessors

The type of scheduling problem most closely related to alternative routing is concerned with 'parallel processors' or 'multiprocessors'. In these problems, more than one processor is available to carry out the same operation, by relaxing the rule stated in 2.3.11, i.e. that there is only one machine available of each machine type.

The interest in multiprocessors is not confined to manufacturing facilities. Analogies may be drawn with packet movement in data communications networks, vehicle control in transportation networks and task assignment in multiprocessor computer systems. These examples all require dynamic solutions, where a controller observes the network and the route chosen depends on the state of the network at the time when a routing decision is required. Dynamic routing is discussed in sections 3.7 and 3.10.

In general three classes of multiprocessor problems have been studied - identical, uniform or unrelated machines. When the machines are identical, the processing time is the same on all machines. When the machines are uniform, the processing times vary in a uniform manner, but if the machines are unrelated, the processing times vary arbitrarily between machines.

The first point to consider when scheduling multiprocessors is whether a job may be divided among 2 or more machines. If not, (batch splitting will not be allowed in this work), then n jobs must be divided into m distinct subsets (where n is the number of jobs and m is the number of machines).

Coffman (18) describes a schedule for multiprocessors as comprising m blocks or subsets, where the tasks in each block are ordered by a permutation to yield the order of task executions for a processor.

Conway et al (11) show that to minimise the flow time, the jobs should be divided among the machines so as to balance the workload as far as possible, using SPT as the dispatching rule, and also balancing the distribution of long and short jobs between machines. Although the simplest equations demand identical machines, the principle of balancing the workload among non-identical machines still holds and processing times on different machines may be viewed as a matrix of n jobs by m machines (fig.8) where p_{ij} gives the time to perform a single operation i on machine j.

A regular measure of performance may still be minimised using the SPT rule. If a job can be divided between 2 or more machines, better schedules are possible (because of the reduced processing time, reduced idle times, reduced waiting times) but determination of the schedules is more difficult. Conway et al prove clear advantages for simultaneous processing ranging from a minimum of 25% reduction in average flow time, for 2 machines, to a maximum of 50%, for many machines, in the idealised, identical machine, identical job condition. Ignoring the penalties of multiple set-up and tooling, Conway et al consider that any schedule can be improved by taking advantage of parallel operations on identical machines.

Figure 8 Parallel machine processing time matrix

		MACHINES				
		1	2	3	...	m
	1	p_{11}	p_{12}	p_{13}	...	p_{1m}
	2	p_{21}				
JOBS	3	p_{31}				
	.					
	.					
	n	p_{n1}				p_{nm}

In the next sections, some of the analytical techniques which have been used to tackle scheduling problems will be described.

It is worth restating that most practical production environments have the following features:

(i) They are dynamic - jobs are continually arriving and moving through the network.

(ii) They are stochastic - although expected operation times are known beforehand, operator or material or machine or tool problems can cause significant variations.

(iii) Good due date achievement and flow time performance is required. Generally a range of performance measures would be used in a production facility including WIP level, due date achievement, lead time, unit cost, quality, machine utilisation.

In addition, investigation of alternative routing demands that rules stated in 2.3.10 and 2.3.11 be relaxed, such that

(iv) Job routing is not necessarily fixed at the start

(v) More than one machine of the same type may be present

2.10 Constructive algorithms

A constructive algorithm builds up an optimal solution from the data of the problem by following a simple set of rules which exactly determine the processing order (19). Many single machine cases are solved constructively, but only a very few problems with two or more machines can be analysed in this way. The two machine, general job shop problem with n jobs and an objective of minimising maximum flow time is the only one for which there exists a constructive algorithm applicable to all cases (11). For all other families, a few constructive algorithms exist that apply only to a few special cases, usually where the processing times are restricted in some way.

Some of the most important cases of constructive algorithms are those due to Johnson (20). In the case of n jobs going through two machines in a flow shop, (i.e. all jobs must go through both machines in the same order, but there may be waiting time between the two operations for any job), an optimal schedule is shown to be constructed by placing jobs having short processing times earlier in the processing sequence for the first machine, (thus minimising delay before the second machine can start), and placing jobs having short processing times later in the sequence for the second machine, (thus minimising idle time of the first machine while waiting for the second machine to finish all its operations). The algorithm is applicable to the general job shop case by dividing the set of n jobs into 4 subsets, according to whether a job is to be processed on one or both machines and in which order. The jobs are sequenced and then combined into an optimal schedule.

A useful perspective was Akers' (21) graphical solution which shows the progress of 2 jobs through m machines on a graph of operations on job 2 (in time x-axis) against operations on job 1 (in time y-axis). It is possible to measure and manipulate idle time by this method and is effectively a permutation schedule. As French (19) remarks, because of its graphical nature, it is a method confined to 2 dimensions and does not generalise in an obvious manner.

2.11 Complete Enumeration

In terms of the sequencing of n jobs through m machines, complete enumeration will require $(n!)^m$ sequences,

e.g. 4 jobs and 1 machine produce 24 schedules

4 jobs and 2 machines produce 576 schedules

4 jobs and 4 machines produce 331776 schedules

Complete enumeration is thus inefficient. Although it is generally assumed that the "optimal" schedule would eventually be discovered, it may be shown that a slightly more complex problem than the one above e.g. $(10!)^{10}$ representing 10 machines and 10 jobs, is impossible to solve before the end of time. This process of identifying and evaluating every possible schedule is also called explicit enumeration. Most approaches to the problem therefore attempt to limit the extent of the search for a schedule and/or generate a sequence from properties of the known available jobs.

2.12 P and NP Problem Classifications

One of the most important means of understanding the real difficulty of scheduling problems has been the recognition of the nature of the complexity of these problems. A problem's "complexity" refers to its execution time on a computer, expressed as a function of the number of bits needed to describe an instance of the problem, e.g. n tasks in scheduling problems, by means of specifying the algorithm and data required. The order of magnitude notation O[ ] concentrates on the terms of a function that dominate its behaviour. If an algorithm has complexity $O[n^2]$, a constant c exists such that the maximum execution time will be cn^2. A sequencing algorithm whose complexity is bounded by a polynomial in n is called a polynomial-time algorithm, and is said to have a polynomial-time solution and be of polynomial-time complexity (18, 19).

Less "efficient" algorithms effectively require a search (at least partial enumeration) of the solution space and have a complexity that is at least exponential in n which require, for example, an algorithm which has complexity O[2n]. It may appear that a problem of complexity $O[n^{10}]$ is no less difficult to manage than a problem with complexity O[2n], but in practice the polynomial orders are generally confined to $O[n^2]$ or $O[n^3]$ (18). Problems with a polynomial time solution are therefore regarded as "tractable", and fall in the P class. Problems without polynomial time complexity fall in the NP class. Since solutions to all P class problems could theoretically be made more complex, they are also potential members of the NP class. Hence the term "NP-complete" is used to describe the members of the NP class for which no P class solution is known to exist.

Many classical, hard problems such as the travelling salesman problem (i.e. in which order should a number of cities be visited to minimise the distance travelled?) and the knapsack problem (i.e. given n items and a limit to capacity or run time, how can the items be divided between the knapsacks/machines to minimise the wasted space or idle time and maximise a measure of performance?) are included in the set of NP-complete problems (sometimes called polynomial complete) (19).

It has been postulated that if a polynomial-time algorithm is found for one of the NP-complete problems, then it is possible to find polynomial-time algorithms for all the others. This theory has been neither disproved nor proved and many mathematicians believe that all NP-complete problems are inherently intractable (18, 19). Almost all sequencing problems stated in complete generality are NP-complete. In other words, polynomial-time solutions have been found for certain one-, two- or three-machine problems, but for most problems of three or more machines it is likely that polynomial-time solutions will not be found although exponential-time solutions may be found.

The implications of this statement on scheduling problem approaches are:

(i) that exponential-time methods such as dynamic programming and branch-and-bound are inherently inefficient and their application depends upon means of limiting the search, or more effectively choosing which area of the solution space to explore further, and

(ii) that constructive algorithms and heuristics constitute a more "efficient" approach in terms of time to find a solution, but may be less "efficient" in their ability to locate the optimum schedule.

2.13 Dynamic programming

Dynamic programming (DP) originates from Bellman (22) and is applicable to many optimising problems. Taha (23) describes DP as a mathematical procedure designed primarily to improve the computational efficiency of solving select mathematical problems by decomposing them into smaller, subproblems. Dynamic programming typically solves the problem in stages, with each stage involving exactly one optimising variable. The computations at the different stages are linked through recursive computations in a manner that yields a feasible optimal solution to the entire problem when the last stage is reached. This process is also called multi-stage programming.

However, the amount of computation is still quite considerable. Conway et al (11) describe the application of dynamic programming to a 5 city travelling salesman problem. To obtain an answer to this problem, 33 smaller problems were solved by reference to 108 terms from the distance matrix or a list of previously solved problems.

Complete enumeration in this problem would have generated (n-1)! or 24 possible paths, each involving 5 terms from the distance matrix i.e. 120 terms. Dynamic programming was still more efficient computationally. For an 8 city problem, DP would have required 16384 terms, whereas complete enumeration would have required 40320. In addition to the time required to manipulate this number of terms, the method is also heavy on storage requirement, since none of the problems at a given stage may be erased until all the problems at the next stage have been solved.

2.14 Branch and bound methods

Branch and bound methods explore "intelligently" the tree of all possible sequences by eliminating certain branches at each stage of the decision. Thus branch and bound techniques are "implicit enumeration" methods. Branch and bound "solutions" dictate the nature of the search; strategies include depth-first and frontier-first.

The number of operations required, and hence the time required to solve a problem by branch and bound is unpredictable whichever search strategy is used (19). Nevertheless, branch and bound does generally perform better than complete enumeration. In theory, this method will always find an optimal solution, but may take prohibitively long to do so. For example, a 10 job, 10 machine general job shop problem set up by Muth and Thompson in 1963 apparently remains unsolved (19).

The extra effort required to calculate good lower bounds at a node, (on which elimination decisions are based, assuming the objective is to minimise a measure of performance), rather than quick but poor ones, has been shown to be worthwhile (24). Since the perceived advantage of the method lies in the ability to produce optimal solutions as quickly as possible, French (19) has suggested that it may be wise to calculate good bounds higher up the tree where many subsequent nodes will be eliminated, and perhaps quick but poor bounds lower down the tree where fewer will be eliminated. Other methods of speeding up the search include priming the search with a good trial schedule, perhaps obtained by heuristics, or accepting sub-optimal solutions. Branch and bound methods have been the most successful of the non-heuristic approaches for solving scheduling problems, but their success depends on the quality of the lower bounds (11, 19) particularly in the early stages of branching. Hence much mathematical theory has considered the design of their determination and needs to be approached with care.

2.15 Integer programming

Under the integer programming approach, the scheduling problem is "recast" as a mathematical programme, to be solved by standard algorithms. French (19) considers that the method is suitable for small problems only since recast scheduling problems can be very large and "empirically this confirms that scheduling problems are in general very difficult and do not just appear to be so."

The method solves mathematical programming problems in which some or all of the variables can assume only non-negative integer values (23) resulting in a mixed or pure integer program. If the objective and constraint functions are linear, the resulting model is an integer linear program. There are 2 categories of linear programming methods:

1. Cutting (or cutting plane) methods - By systematically adding special "secondary" constraints, the solution space is gradually modified until a point is reached which satisfies the integer conditions. Thus certain parts of the solution space containing infeasible integer points are effectively cut out.

2. Search methods - "Clever" tests are developed that consider only a small portion of the feasible integers explicitly but account for the remaining points implicitly. The most prominent search technique is branch and bound.

Conway et al (11) described one problem of 4 machines and 10 jobs which required 220 variables and 390 restraint equations. They proposed that researchers might have been discouraged from integer programming by the size of the resultant problems and the difficult behaviour of contemporary integer programming computer codes, which together made integer programming a costly method.

Algorithms which have been developed for the integer problems have not been found to be uniformly efficient in terms of computer effort, particularly as the size of the problem increases. The performance of integer algorithms is therefore considered to be erratic (19,23) largely due to rounding errors.

2.16 Critical route analysis

Critical route analysis (such as the well-known CPM and PERT methods) is used to plan projects or jobs which may be broken down into tasks or jobs of single operations. The operations must be capable of being ordered such that the independencies and inter-dependencies may be represented (precedence between tasks/operations). The aim is to minimise makespan i.e. the time between start and terminal nodes.

Critical route analysis assumes that an operation may start as soon as the preceding operation(s) have been completed. Waiting time is caused by delays due to limited resources (even though all precedence constraints have been satisfied) and by delays of operations which cannot start until precedence constraints have been satisfied. The delays due to limited resources are more difficult and it is usually assumed that there is an unlimited supply of machines. Queuing delays are therefore non-existent, although there are delays due to precedence restrictions.

The criticality of an operation is determined by the latest time at which it may be completed without increasing the completion time of any subsequent operations or the overall project/job. The particular usefulness of critical route analysis is the determination of the shortest route between the initial and terminal nodes. According to French (19), although the algorithm is simple, the ability to process an unlimited number of tasks simultaneously is very unrealistic in practice, and a variety of problems have been studied in which limits on availability of machines are imposed. All are NP-hard.

2.17 Multiple server queue theory

It is reasonable that job shops and flow shops could be treated as queuing networks in which there are arrangements of multiple servers (different machines) with customers (jobs) visiting more than one server before discharge from the system. Early work (25) concentrated on identifying the special conditions under which the machines in the network could be treated independently and where the individual queues could be analysed separately. These conditions were:

1. The input process must be Poisson
2. The routing of a job i.e. the determination of which machines are to be visited and in what order, is entirely independent of the state of the system.
3. The processing times are to be exponentially distributed.
4. The order in which jobs are sequenced on a particular machine must be independent of

   (a) the processing times

(b) the subsequent routing of the job

(c) the knowledge of specific future job arrivals to the machine

Conway et al found the 4th condition to be particularly restrictive since it precluded comparison of sequencing procedures in queuing networks. Work on conditions for alternative routing was effectively precluded by the second condition. Theoretical analysis of systems where any of these conditions was relaxed became extremely difficult, but assuming the stated conditions were met, arrangements equivalent to job shops, flow shops and identical multiple servers have been investigated.

Queuing theory is used principally to investigate dynamic and stochastic scheduling i.e. jobs continue to arrive, and processing times are uncertain but conform to a known distribution. Although scheduling problems can be difficult to solve it is possible to test the effect of different priority rules or service disciplines on single independent machines and hence equivalent priority rules to the static, deterministic case have been tested and compared. Panwalker and Iskander (15) include results of the performance of these priority rules in their survey.

A useful proof (23) using queuing theory shows that a pool of servers reduces the waiting time that would be incurred by customers/jobs waiting for the same number of independent servers, though the utilisation level would be the same in both cases. This outcome is pertinent to groups of similar machines. Hence, a pooled buffer stock should produce shorter waiting times than dispersed buffers.

A number of queuing network models have been used successfully to check preliminary design values for flexible manufacturing systems (FMSs). Suri (26), co-author of one such model and Marchal (27) have compared queuing network models with other evaluative models for FMSs and conclude that they are quick to use but produce "ballpark" (approximate) decisions.

Seidmann et al (28) review the capabilities of several queuing network models which are used to study gross tradeoffs between principal design parameters during FMS preliminary design stages. They examined 4 analytic queuing models and one simulation model written as a generic model in the GPSS language. The 4 analytic models routed parts between machines using a routing probability matrix comprising the probabilities that a particular part would visit a particular machine. The simulation model used a fixed routing matrix. Each workstation in all the models reviewed comprised 1 or more parallel identical servers which are all fed by a common buffer, exploiting the observation of reduced waiting time described earlier. All the models examined were designed to investigate how plant utilisation and production rate were affected by design and control variables such as the number of transporters in the system, process plans, the number of pallets and priority rules used to despatch parts, and did not appear to address alternative routing mechanisms.

2.18 Simulation as an experimental technique

Simulation has been used since the early fifties as an alternative approach to algebraic and probabilistic methods for investigating scheduling problems. Conway et al (11) classified most studies as either:

(i) investigations which attempted to extend theoretical results, or

(ii) attempts to solve actual problems in real shops before installation.

They observed that there had been few publications of the latter type and those they had seen presented "no great surprises". In the succeeding twenty years, the role of simulation in manufacturing has changed dramatically and indeed the number of publications of the second type has increased manyfold (e.g. 29, 30, 31, 32). The increase in use of simulation has arisen primarily from the advances in computer technology which have allowed larger, faster, and more complex models, and subsequently from the development of visual, interactive computer simulation systems (33, 34) which have improved communication between operations researchers and domain experts. Many commercial packages are now available in Britain, and the trend among these products is to reduce the operations research background and programming effort required to build simulation models.

A survey undertaken by Kiran and Smith (35) highlighted some of the modelling problems faced by researchers using simulation to advance theoretical knowledge of job shop behaviour. A brief summary of the main sections shows the variety of conditions and assumptions. In general, simulation is probably the technique which has been most used to investigate dynamic and stochastic environments (19).

2.18.1 Arrivals.

Three ways of modelling arrivals were used in the papers surveyed:

(i) instantaneous release of the job into the shop on arrival (most common)

(ii) periodic release of all available jobs at the beginning of the period

(iii) job pooling i.e. a subset of available jobs is released periodically.

The most popular arrival pattern used a Poisson process, but others used one of a variety of distributions e.g. Erlang, uniform, geometric, bimodal, binomial.

2.18.2 Set-up and processing times

A variety of distributions to generate processing times had been observed, and it was considered that the distribution used affected the shop performance. In addition, some priority rules were more sensitive to processing time distributions than others. For example, one source (36) observed that the performance of non-due date scheduling rules improved as the variability in processing time decreased. Apart from a few studies, set up times were assumed to be included in the processing times, and in some cases were taken as a function of the processing time.

2.18.3 Number of machines and job routing

Researchers who had investigated the effect of machine shop size on the results of priority rule investigations found that there was no significant effect. A 4 machine shop is large enough to represent more complex shops.

Kiran and Smith (35) describe level 2 and level 3 type alternative routing modelling options (sections 1.6.2 and 1.6.3) without reference to papers surveyed (if any). They also state that alternative routing has a significant impact on shop performance and relative effectiveness of priority rules, providing better performance and reducing the difference between priority rules, but again without reference to experimental work.

2.18.4 Due dates

Shop performance and relative effectiveness of priority rules are affected by the method of due date assignment, as well as by the tightness of due dates.

2.18.5 Performance criteria

Three categories are offered:

(i) criteria based on job completion times, in-process inventory and utilisation

(ii) criteria based on due dates

(iii) criteria based on costs

The remaining sections of the survey describe the variety of priority rules observed and their relative performance against the different criteria.

In his review of evaluative models for flexible manufacturing systems (FMSs), Suri (26) first distinguishes "evaluative" (descriptive) models from "generative" (prescriptive) models. Generative models find 'good' candidate decisions whereas evaluative models evaluate a given set of decisions. Simulation is firmly in the evaluative category, providing insight into the way that a system handles decisions, but has been criticised for the time that may be required to find a decision which is 'good' (26). Suri (26) also states that simulation models can be very accurate, although the time required to create the model, to generate detailed data sets and to carry out a number of runs can be expensive. Suri recommends different evaluative models to address different questions in FMS design. Despite its drawbacks, simulation is recommended for most of these questions, including decisions about routing.

2.19 Heuristics in scheduling

2.19.1 Properties of heuristics

Heuristics have been used in scheduling problems because of their speed, visibility and comprehensibility. Although heuristics rarely yield optimal solutions, they are more likely to be installed in practice because of their ease of use (37).

Some guidance on the definition and use of heuristics has been provided by Silver, Vidal and de Werra (37). They quote Nicholson (38), who defined a heuristic method as a procedure "for solving problems by an intuitive approach in which the structure of the problem can be interpreted and exploited intelligently to obtain a reasonable solution".

Silver et al recognised that realistic formulations of complicated decision problems are likely to lead to mathematical problems which are very difficult, if not impossible, to solve exactly. Approximate solution procedures are therefore important. In this category of difficulty, they choose the example of large NP-complete problems for which it appears that efficient optimal solutions may not be possible.

Four properties of a good heuristic are listed:

(i) Realistic computational effort to obtain the solution.

(ii) The solution should be close to the optimum on average.

(iii) The chance of a very poor solution should be low.

(iv) The heuristic should be understandable by the user, preferably explainable in intuitive terms.

2.19.2 Gere's early heuristics

Heuristic scheduling methods attempt to duplicate or better the performance of skilful schedulers. Loading rules, or priority rules can be regarded as heuristic rules which are not necessarily very skilful.

Gere (17) introduced some important scheduling heuristics, which were combined with a number of priority rules. Their effectiveness was tested under static and dynamic conditions. The heuristics included the following procedures:

2.19.2.1 "Alternate Operation"

* schedule the operation according to the priority rule
* check to see if this makes another job "critical"
* if so, revoke this scheduled operation and instead schedule the next operation on the critical job

* check again for lateness

* if the second operation does not cause any job to be critical, then leave it scheduled, else schedule the first one (which had been dictated by the rule)

2.19.2.2 "Look Ahead"

* when an operation has been scheduled, find out if there is a critical (i.e. late, or nearly late) job due to reach this machine at some future hour, but before the scheduled operation would be completed

* if so, schedule the critical job

* check the effect of this on other jobs

* compare lateness caused to other jobs by scheduling the critical job with lateness caused by scheduling the first job and choose the one with least effect.

2.19.2.3 "Insert"

* if a "look ahead" job has been scheduled, there is a period of idle time on the machine

* try to find a job whose next operation may be completed before the look ahead job is due to arrive.

Clearly, using the "insert" rule with the "look ahead" rule is more effective than using the look ahead rule without it. Gere found the alternate operation and look ahead rules to be effective since they improved the performance of all the priority rules tested. An increase in computing time of only 10-20% of the total was observed. Conway et al were critical of Gere's heuristics (and presumably most other heuristics) for their inability to contribute to the development of scheduling theory.

2.20 Conclusions

Mathematical approaches are becoming increasingly imaginative, varied and intricate in their attempts to find optimal solutions to scheduling problems. The recognition of the intrinsic complexity of even idealised examples of scheduling environments validates alternative approaches such as heuristics which may yield solutions to an acceptable performance, even if they are not optimal.

The dynamic and stochastic characteristics of real factories allow investigation by few methods, of which simulation is the most flexible. It is to be hoped that mathematical approaches will eventually reach a stage of development where they may be combined more readily into factory control systems, and be more widely understood by production schedulers.

The methods described in this chapter have been largely unable to address alternative routes because of the resulting mathematical complexity. Simulation has been the most appropriate investigative technique. A summary of work on routing flexibility is presented in the next chapter.

3. Routing Flexibility

This section examines previous approaches to routing flexibility, starting with early work by Russo, Wayson and Neimeier. More recent work on routing flexibility in flexible manufacturing systems is then examined, followed by a discussion of rules used to achieve flexible routing in packet-switched communications networks. Finally, some further examples of attempts to describe and achieve routing flexibility are presented.

3.1 Russo

In the introduction, Russo's (3) approach to alternative machine-operation ordered pairs was described using partial routes. If an operation was a member of a partial route of N operations, each operation in the partial route would be considered to have N-1 alternates. The choice depended on the queue at the machine where each alternate operation could be processed. Russo used a simulation model to test the effect of different queue assignment rules at the transitional level (level 2) and also examined some despatching rules at the operational level (level 3). Performance was measured by mean tardiness and mean wait time in comparison with mean tardiness and mean wait time in a run where the queue was assigned randomly. The four transition level heuristics to assign jobs to queues were:

(i) LOAP - lowest average priority (used in conjunction with a selection of despatching rules which were used to assign the priority)

(ii) MASRO - maximum average slack per remaining operation

(iii) MACSD - maximum average critical start date

(iv) MAPT - maximum average processing time (which was expected to be analogous to the shortest processing time despatching rule because queues of longer tasks would be chosen, rather than joining a queue of shorter tasks where the new job would compete for time with short operations).

Russo found that the slack based rules (MASRO and MACSD) performed better with regard to reducing mean tardiness but had little effect on wait time. MAPT had little effect on tardiness or wait time. By the very nature of the partial routes used and the way in which all processing times and routes through the theoretical job-shop were chosen, there was no inherent advantage in altering the order of operations within the partial route. All the jobs had to go through all the machines at some time anyway.

There was some advantage in choosing alternative routes at the operational level, where a job was entered into all the queues where it might be processed next. On loading to one machine, the job is removed from all queues. This method is clearly advantageous and is analogous in practice to machines drawing work from a common buffer.

3.2 Wayson

In contrast, Wayson (4) found a significant advantage in being able to process through alternative routes. The "schedules" in figure 2 show the availability of a particular alternate machine for any machine in a 9 machine shop.

On entry into the system or on completion of a non-terminal operation, the specified machine for the next operation is found from some previously determined route. Using the alternate machine availability matrix, the specified machine is used as a key to generate a set of alternatives. The job is then routed to the alternative machine which has the shortest queue of jobs.

In the case of a tie between the specified machine and an alternate, the specified machine would be chosen. In the case of a tie between alternates, one would be chosen at random.

The model comprised a 9 machine shop, with no breakdowns, tool failures or other unplanned stoppages. Job inter-arrival times and processing times were sampled from exponential distributions, whose means were chosen to obtain an overall machine utilisation of 90%. Routings were random and averaged 9 operations per job. Two dispatching rules were used to choose jobs from queues awaiting processing, FCFS and SPT. Four measures of performance were used:

(i) average number of jobs in the queue

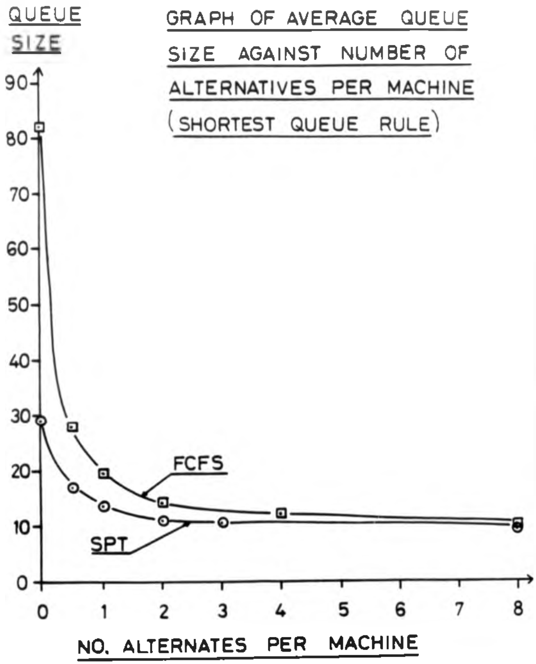
(ii) average idle time

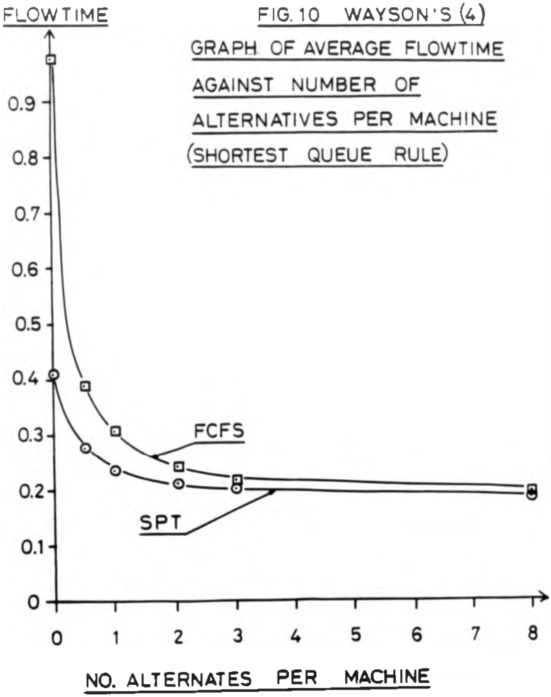
(iii) average flowtime of all jobs completed in a given interval

(iv) average proportion of time that an alternate was selected in preference to the specified machine in relation to the number of times that this decision was made in the decision interval.

Figures 9 and 10 show that the queue size and flow times fall dramatically if one or two alternate routes exist. Thereafter, there is little further benefit. Wayson concluded that the shortest queue heuristic had a powerful effect of maintaining even workloads throughout the shop, even when alternates were

FIG. 9 WAYSON'S (4) GRAPH OF AVERAGE QUEUE SIZE AGAINST NUMBER OF ALTERNATIVES PER MACHINE (SHORTEST QUEUE RULE)




FLOWTIME
FIG. 10 WAYSON'S (4)
GRAPH OF AVERAGE FLOWTIME
AGAINST NUMBER OF
ALTERNATIVES PER MACHINE
(SHORTEST QUEUE RULE)
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
FCFS
SPT
0
1
2
3
4
5
6
7
8
NO. ALTERNATES PER MACHINE

available for only half the time. The number of times that an alternate was chosen did not increase linearly with the number of alternates available - fig.11. This was attributed to the increased number of ties, which were resolved by assigning the job to the specified machine after all.

Perhaps the most important finding was that the behaviour of the SPT rule under unique machine assignments was not very different from performance using the FCFS rule when alternative machines were available for at least 20% of the decisions. The difference in performance between SPT and FCFS decreased as the number of alternates increased.

3.3. Neimeier

Neimeier (39) was aware of Wayson's work (they were both working under Conway and Maxwell at Cornell) but did not appear to be aware of Russo's work under Carroll at MIT. Neimeier follows the definition of alternate routing stated by Russo where the order of operations on certain partial routes may be shuffled, but every operation must be performed on its specified machine (i.e. no alternate machine availability). A precedence diagram (fig.12) may be drawn to show the relationship between operations. Within a group, the operation to be carried out next is chosen by examining the queues at each of the possible work centres, and choosing the work centre which had the shortest queue.

Because all originally routed operations must eventually be performed on their specified machines, Neimeier suspected that his version of alternative routing would not perform as well as Wayson's alternate machine availability. He foresaw a condition where a long queue on one machine may not dissipate while

FIG.11 PERCENTAGE OF DECISIONS WHICH SELECTED AN ALTERNATE MACHINE AGAINST NUMBER OF ALTERNATES/MACHINE (WAYSON,4)

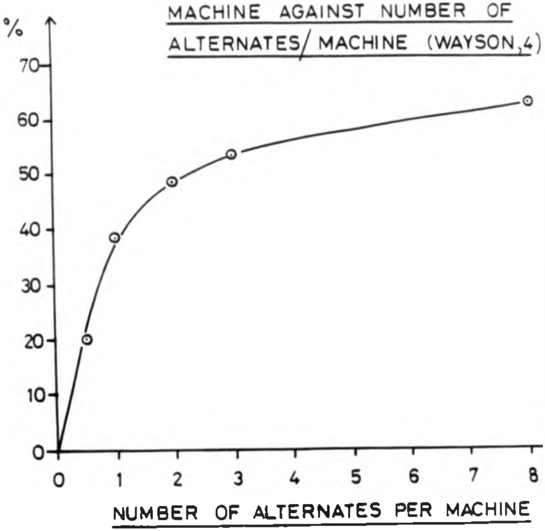
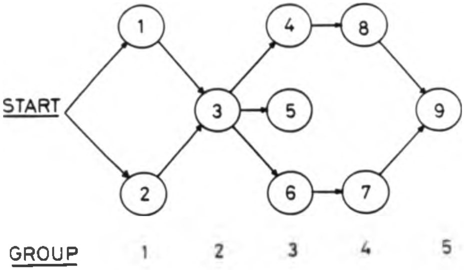


FIG. 12 USING A PRECEDENCE DIAGRAM TO ILLUSTRATE ALTERNATIVE OPERATIONS

(NEIMEIER,39)



- CIRCLED NUMBERS ARE OPERATIONS.
- ALL OPERATIONS IN A GROUP MUST BE COMPLETED BEFORE OPERATIONS IN THE NEXT GROUP MAY START.

the alternate operations were being carried out, and a job would still have to join that queue because it was the last operation in the group, (and must be performed on the specified machine). To gauge the effect of increasing flexibility in routing, routes were generated with groups of increasing average size.

Neimeier used the same job shop model as Wayson. Four measures of performance were used:

(i) average number of jobs in the queues of all machines

(ii) average flow time per job

(iii) average idletime for all machines

(iv) proportion of operations that are performed out of sequence.

Flowtimes and queue sizes were observed to fall as the number of alternates selected increased - using both dispatching rules (figs. 13 and 14). Again, more dramatic effects were observed using the FCFS priority rule. Alternate routing reduced the variability in queue size with a consequent balance of machine workload. In the event of a tie, the job joined the queue of the originally routed operation.

Of more interest are some extra runs that Neimeier performed on an "unbalanced" job shop i.e. mean operation times vary between machines. Results from two sets of operation times 0.92-0.96-1.0. and 0.88-0.96-1.04, indicated that the greater the degree of unbalance in a job shop, the poorer will be the performance improvement obtained by the use of alternate routing.

FIG.13 AVERAGE FLOWTIME AGAINST PROPORTION OF ALTERNATIVES SELECTED (NEIMEIER, 39)

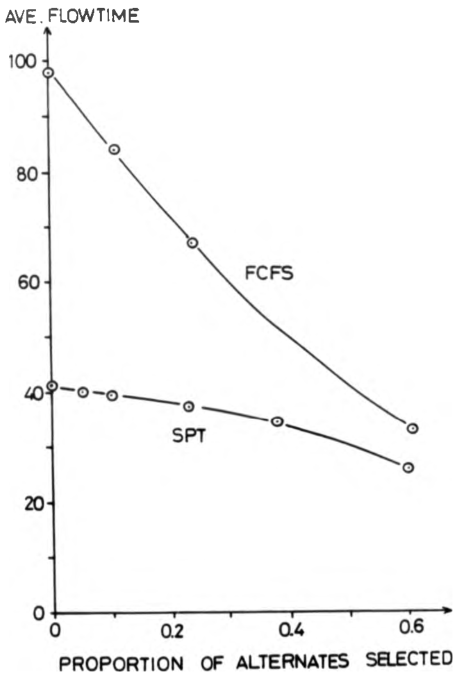
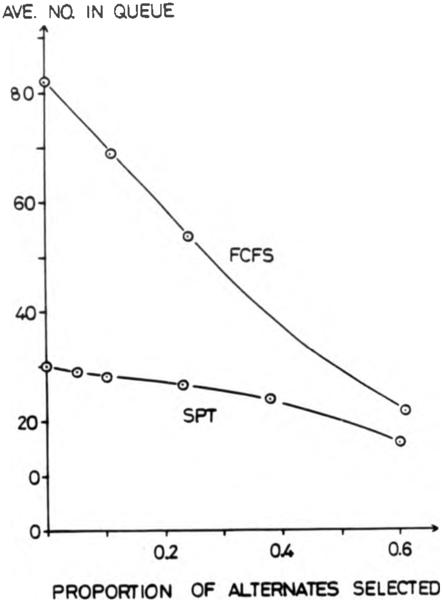


FIG. 14 AVERAGE NUMBER IN QUEUE AGAINST PROPORTION OF ALTERNATES SELECTED

(NEIMEIER, 39)



The variation between these operation times is not great and brings into doubt the whole question of whether alternate routing is worthwhile at all in practice when it is known to be difficult to manage. Conway, Maxwell and Miller were emphatic in their endorsement of alternative routing with regard to Wayson's work. They stated that in terms of practical implementation of scheduling procedures, the improvement was too important to be neglected. A sophisticated scheduling procedure that did not take advantage of this type of flexibility would risk being outperformed by a knowledgeable human scheduler.

3.4 Route Flexibility in Flexible Manufacturing Systems (FMS)

In a joint paper (40) several workers in the FMS field unified their definitions of "flexibility" in flexible manufacturing systems. Following the definition of FMS as an "integrated, computer-controlled complex of automated material handling devices and numerically controlled (NC) machine tools that can simultaneously process medium sized volumes of a variety of part types", 8 types of flexibility were defined:

* machine - changes required to produce a given set of part types
* process - ability to produce a product mix, without batching
* product - ability to accommodate a new product
* routing - ability to continue production despite breakdowns
* volume - ability to operate profitably at different volumes
* expansion - capability of accepting modular expansion
* operation - ability to interchange ordering of several operations

* production - the "universe" of part types that may be produced, taking into account earlier flexibility types.

3.5 Routing Flexibility

Routing flexibility was considered to exist (40) if either (i) a part type can be processed via several routes or (ii) if each operation can be performed on more than one machine. Flexibility in scheduling can therefore be (i) potential, where part routes are fixed, but parts are automatically rerouted when a breakdown occurs, or (ii) actual, where identical parts are actually processed through different routes, independent of breakdowns. It was suggested that routing flexibility can be measured by the "robustness" of the FMS when breakdowns occur i.e. the production rate does not decrease dramatically and parts continue to be processed. Falkner (41) comments that this implies that routing flexibility would appear to be achieved if there are at least 2 machines in the system which are capable of processing each operation type. Falkner observes that the overall production rate following a breakdown will depend on the load scheduled for the alternative processes. If they are heavily loaded, the planned schedule may not be achieved. Thus, some planned under-utilisation would be necessary in order to maintain the required production rate.

Three ways of attaining routing flexibility were listed by Browne et al (40):

* allowing for automated and automatic rerouting of parts (potential routing flexibility)
* pooling machines into machine groups (actual routing flexibility)
* duplicating operation assignments (actual routing flexibility).

Work on dynamic scheduling and routing in FMSs is changing automatic routing of parts from potential to actual routing flexibility, and will be discussed in section 3.7. Machine pooling is considered in section 3.9. Duplicating operation assignments aims to create identical machine capabilities, subject to tooling constraints. Tool management will not be considered here but rules for routing through identical and non-identical machines will be discussed later.

3.6 Operation Flexibility

Operation flexibility is significant because it challenges the rigidity of the classical fixed operational order specified by a process planner. Although there is usually some necessary partial precedence structure for a particular set of operations, the sequence is rarely so sensitive that no order changes can be made. This flexibility has a similar effect to the partial routes considered by Russo (3) and Neimeier(39).

The difficulties of managing operation precedence using conventional computing capability are daunting, but operation flexibility is a reality of emerging knowledge-based scheduling systems (42,43).

3.7 Dynamic Routing in FMS

A variety of methods have been used to effect dynamic routing. Maimon and Choong (44) developed a dynamic routing policy to achieve real-time load balancing across groups of machines in order to increase throughput and reduce work-in-progress (WIP) queues. The routing policy measures the input WIP each period in terms of processing time to each group of machines, and determines which groups should send jobs to, or receive jobs from, other groups of machines. A dynamic programming approach was used to make this decision, using a variant of the (s,S) inventory control policy, where a stock replenishment order is placed at a periodic review only if the minimum stock level s has been breached. The quantity ordered is the difference between the stock level at the time of review and some maximum level S. Hence the routing policy is effectively fixed until the next periodic review.

Cassandras (45) developed a 3-level hierarchical control system for a dynamic handling system for FMS that chooses the shortest time path. An assessment of the alternative routes available is made on arrival of a job into the system. A middle level "control coordinator" determines the route and informs the lowest level nodes of the route for that job. Although the materials handling network is now aware of all routes for all current jobs and is therefore resilient to a coordinator breakdown, fixing the routes on arrival may mean that the routes are no longer truly dynamic, i.e. they are unable to change further according to changing local conditions.

Yao and Buzacott (46) tested a randomised version of the (deterministic) "join the shortest queue" rule, called probabilistic shortest queue (PSQ). Parts are routed with the highest probability to the workstation which has the largest number of empty spaces or, alternatively, the relatively shortest queue. PSQ was adopted because it could be modelled mathematically and because it had some impact on reducing blocking and increasing machine utilisation. Routing decisions are made when a transporter becomes available at the entry point to load a new part into the system. The choice of parts made ready to be loaded is considered to be a higher-order control problem and is not addressed. When a decision is required, PSQ merely determines to which workstation the next part should be delivered according to the distribution of frequencies allocated to each alternate. The algorithm performed well compared with fixed probability routing. Computer execution time for very simple examples (2 part types) was of the order of several minutes and Yao and Buzacott suggest that approximations would be required to reduce the demand on memory and computing time for a large variety of part types.

Another strategy that made heavy demands on computer time was tested by Hahne (47). Although she did not specifically refer to FMS dynamic routing, it is appropriate to consider her work here.

Hahne tested an optimal routing strategy on a simple arrangement of 3 unreliable machines (one lead machine feeding 2 others) and finite storage buffers, modelled as a Markovian decision process. If the buffer of one downstream machine was full, or one downstream machine was broken down, the lead machine continued to process into the alternative buffer. The optimal strategy, using a successive approximation algorithm, demanded a large computation time, taking an hour to make a decision and working to 30 decimal places. A heuristic was developed to balance the workload between the 2 downstream machines. The heuristic was tested on the network with perfectly reliable machines and was found to perform as well as the optimal strategy, to within 3 decimal places. Unfortunately this heuristic was not tested on unreliable machines. Hahne admits that the optimal strategy would be unworkable on more complex networks and proposes that further development should include heuristics.

A different approach is taken by Whitt (48), who suggests that an approximate analysis of a more exact model might be more fruitful than these rather exact analyses of approximate models. A package called Queuing Network Analyzer (QNA) was developed to calculate approximate congestion measures for networks of queues, originally in packet-switched communications networks. A version for analysing manufacturing lines was launched later (49), capable of representing machine breakdown and different batch sizes, allowing non-exponential service-time distributions and non-Poisson arrival processes. However, only deterministic or totally random routes were allowed through the network.

3.8 Real-time Scheduling in FMS

It is difficult to draw the line between real-time scheduling, where jobs are allocated to time slots on machines within a very short time period, and dynamic routing where jobs are routed to a machine, chosen from among those machines which are capable of the next operation. It might appear from the following work that despatching rules continue to be important in real-time scheduling, whereas routing decision rules are of primary importance in dynamic routing policies.

Chang et al (50) presented a two-part method for real-time scheduling in FMS. They argued that local despatching rules do not take advantage of global information. In the first part, many feasible schedules are created for n available jobs based on the outcomes of N previously scheduled jobs. Enough schedules are created in order to be confident that the optimal or near-optimal schedule is included. The second stage uses an algorithm to choose the optimal schedule from among those created. In terms of mean flowtime, the method is reported to perform better than SPT, LPT, FCFS, MWKR (most work remaining) and LWKR (least work remaining) rules.

Svestka (51) prefers to discuss real time re-scheduling, where dynamic dispatching rules use current system status as input information at the time that rescheduling is required. An interesting table is included, showing the improvement in performance of a despatching rule as more information is used in the decision, for 10 common despatching rules. The "Real Time Rescheduler" is shown to perform well against an optimal predefined schedule.

3.9 Machine Grouping/Pooling

While examining loading and control policies for FMS, Stecke and Solberg (52) found that classical job shop scheduling theory could not be applied to flexible manufacturing systems. Indeed, the "shortest processing time" (SPT) priority rule performed below average in this work when used alone.

A number of different "loading" policies were tested i.e. assignment of operations to a particular machine or group, ranging from "assign each operation to only one machine and then balance the workload (fixed route)" to "assign each operation to a group/pool of capable, like machines, where all machines are grouped by capability". One further loading rule attempted to minimise the part movement between machines by assigning consecutive operations to one machine as far as possible. Sixteen priority rules were tested to examine scheduling effects in combination with the loading rules.

The maximum machine pooling and minimum machine movement loading rules outperformed the other loading rules as measured by system output level. Maximum pooling performed best in combination with a priority rule called "SPT/TOT" i.e. load the job which has the lowest value of the ratio found by dividing the shortest processing time for the operation by the total processing time for the job. This rule is claimed to take account of the properties of each operation as well as each part type. The minimum movement result had been unexpected because the loading had appeared to be unbalanced. Using this loading rule, the priority rule appeared to have little effect on the output.

3.10 Packet switching communications networks

Routing of jobs through networks of machines is just one example of a routing problem. Examples of routing and resource assignment problems are found in other contexts, including the control of vehicles in transportation networks and the routing of packets in data communications networks.

In a packet-switched computer communications network, routing is the term used to describe the decision-making process by which a given node selects one or more of its outgoing lines on which to forward a packet which is on its way to some ultimate destination (53). Important factors in routing policy are the frequency of updating lists of available nodes, and the algorithms used to make decisions at the nodes.

Routing control methods may be broadly classified as static, quasi-static or dynamic (54, 55) according to the frequency at which they are updated. Under static control, given proportions of the traffic coming into any node are allocated to each outgoing line. The proportions are determined by calculation or from long term performance of the network and are fixed before the network starts operating (54). This policy is simple to implement, but does not respond to failure of a link or node where a message could be completely blocked, or to heavy congestion in one region of the network (55).

Quasi-static control allows periodic routing changes at given intervals or when extreme situations occur. After updating, the routing decisions are consistent until the next update. Although information is periodically received, information may not be received from a failed link or node because it has failed (54). It might be argued therefore that that lack of information should indicate failure and that the important characteristic is the updating interval, which will determine currency. It is not clear whether a blocked packet may be diverted if it is realised that a particular link or node has failed.

In dynamic routing, the next stage of the route is determined according to the state of knowledge at the current node (54). Completely dynamic routing attempts to counter congestion and failure problems, but can place heavy demands on transmission capacity (the very resource it is trying to manage) for purposes of readdressing and reordering messages (55).

To assist the dynamic routing decision-making process at the node, algorithms have been sought which are easy to calculate, are adaptive to changing levels of traffic flow and to conditions in the network (e.g. failure), are fair to all packets and are able to provide the "best" route that minimises packet delay and maximises throughput (53).

These routing algorithms can be classified as adaptive or non-adaptive. Non-adaptive routing algorithms make decisions according to some pre-determined rule (and are merely local manifestations of static control). Adaptive algorithms can respond to changing traffic and network conditions by taking packet delays, line utilisations and line operational status into account (53), and can therefore enable truly dynamic routing.

Non-adaptive routing algorithms which are feasible for manufacturing applications include:

(i) Random walk, which chooses one of the available outgoing lines at random (each line has an equal chance)

(ii) Fixed directory routing, which maintains a list of lines allocated to a particular destination

(iii) Split traffic routing (also known as directory routing or traffic bifurcation) which lists one or more outgoing lines to each destination. Each line has an associated probability of being chosen. Arrival of a packet at the node causes a random number to be selected and a line is chosen from the list.

Adaptive routing algorithms are further classified as centralised, isolated or distributed. Networks using centralised routing are controlled by one routing control centre (RCC) which contains routing tables of all nodes and the allowable lines to their destinations. The nodes pass status information to the RCCs, such as delays, and queue lengths (53, 56). This method of operation has some potential in manufacturing if nodal (machine) capabilities become more general and computer-integrated manufacturing becomes a reality, but manufacturing equipment is unlikely to reach the same level of commonality as communications nodes.

In isolated adaptive routing, each node takes account of local network conditions, based on information it has gathered itself and without transmitting routing information to other nodes. Isolated adaptive algorithms include (53, 56):

(i) Isolated shortest queue (also known as "hot potato" routing) where the packet is added to the line which has the shortest outgoing queue, without regard to the line's destination. Split traffic routing is usually used to break ties.

(ii) Backward learning technique, where packets return progress information from each node through which they pass to all earlier nodes through which they travelled, so that nodes may "learn" the distances and choose the outgoing line that achieves the shortest distance to a given destination.

In distributed routing, each node collects information about delays and queue lengths from its neighbours, and also maintains a routing table of the preferred outgoing line for each destination, based on minimum delay or some other measure.

Bell and Jabbour (56) evaluated random walk, fixed directory, split traffic, isolated shortest queue and backward learning algorithms by simulation. Random walk showed the longest delays since no preference is given to better or shorter routes. Isolated shortest queue substantially reduced the queuing time and hence the delays. Split traffic showed far smaller delays, due to the strong preference given to the shortest path. However, fixed directory and backward learning showed the smallest delays since both algorithms implement true shortest path routing.

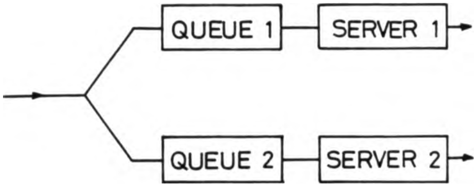
3.11 State Dependent or Dynamic Routing in Communications Networks

State dependent or dynamic routing refers specifically to decision rules which take the state of the network into account at the time the decision is made.

Ephremides et al (57) studied the simple network shown in figure 15. They showed that if the number of jobs already in each queue is known at each decision time, the optimal decision is to route jobs to the shortest queue. However, if the length of each queue is not known, then it is best to alternate jobs between queues (the "round robin" rule), provided that the queue lengths are equal initially.

These conclusions may appear to be intuitively obvious but Foschini and Salz (58) use diffusion theory to yield an approximate solution to a similar problem and comment that "A rigorous reckoning ..... and the proof ..... are formidable convergence problems which are annoying obstacles for one interested in applications. A difficult digression into a highly specialised area is needed just to rigorously affirm an item that in retrospect is likely to be treated as intuitively obvious." Foschini and Salz also comment that the basic alternate routing problem (as in fig.15) had remained unsolved by queuing theory.

FIG. 15 SIMPLE COMPONENT OF A COMMUNICATIONS NETWORK STUDIED BY EPHREMIDES ET AL (57)



(IDENTICAL, INDEPENDENT, EXPONENTIAL SERVERS)

3.12 Production Decision Support System

To take advantage of the flexibility offered by FMS, fixed operation orders and fixed routes, which have been the basis of most scheduling theory, are no longer viable assumptions. Important progress has been made on new approaches to scheduling under new conditions. These issues have been forced to the forefront of study because of the high capital investment in FMS, demanding high utilisation, but allowing acquisition and manipulation of data to make decisions which had previously been infeasible to contemplate in practice.

Several authors have restated the increase in productivity that routing/operation flexibility offers (59, 60, 61). Chen and Talavage (59) describe the case of potential routing flexibility as an "unstructured problem" which requires the unavoidable judgement of an experienced production controller. This decision-maker is given data via a production decision support system from which alternatives can be established, and which are to be evaluated from "experience", possibly by experimentation with simulation on various alternatives. No guidance is offered on how alternatives may be constructed or evaluated.

Nof et al (60) showed that part-mix (selection of which part types to be processed and their relative volumes) has a significant effect on productivity. The two other areas of operation control for FMS were listed as selecting the process from alternatives available and the sequence of loading and advancing parts. Results showed that sequencing also had a significant effect on performance but regarding alternate routes, the results were less clear. It was suggested that the system should be presented with "a range of alternative process plans for each of a range of part types and let some algorithm select the parts and processes and determine the appropriate mix ratios." No algorithms to choose from alternatives were offered.

3.13 Control Software Structure

In their modular, generic software system for FMS control, Manara et al (62) describe an interesting and clear method for handling the complexities of partial routes and alternative routes. The topology of a plant is represented by a "graph" (network) whose nodes are associated with either a physical element e.g. machine (physical node), or a functional element e.g. decision "parent node" of alternative operations described later (virtual node). Nodes are connected by arcs which represent the possibility that a part can pass from one node to another. Various node types are described, including the following.

A "virtual parallel" node is a set of interchangeable physical nodes for a given operation. The node may be further described as "symmetrical" where it is a parent node joined to a set of identical children nodes, any one of which may be randomly assigned to the same operation, or asymmetrical where the node is the parent of non-identical children.

Another node type is a "virtual-auxiliary" where the node is parent to a set of child virtual nodes, which is common to some paths in the network, and/or may occur more than once in the same path (e.g. washing, duburring, handling robot).

Thus the database includes:

(i) node description library relating node identity (id), type, state (available, non-functioning etc.) and links to other nodes

(ii) operation parameters library which lists operations which may be carried out by a node and the associated parameters (tools, set up time etc.)

(iii) production schedule which lists batches to be processed and their features (id, quantity etc.)

The main purpose of this work was to describe the software structure and no indication is given as to how well it is working at the two major installations described, or indeed, how a route is finally chosen from among the alternates available.

3.14 Evaluation Heuristic

Schlauch (61) addressed the problem of selecting the "best follower" order from a list of options by establishing a "coefficient" for each order. The options were established by following a number of boundary conditions which included alternative machines, alternative sequences of operations, and alternative operation variants. This complicated method evaluates a coefficient, which measures the ability of the alternative to minimise idle time, for each alternative for each operation. The coefficients for each operation are combined together to form a coefficient for each order, and compared with the coefficients calculated for alternative operations sequences. Results showed high machine utilisations were achieved.

3.15 Process industry example

A simple case of alternative routing strategy in a process industry is described by Grinsted (63) where jobs must be allocated to 1 of 8 different 2-station lines. After processing through the line once, jobs may be recycled through the lines for between 1 and 4 further stages of processing. About 70% of the jobs pass through one particular line for the first cycle. The problem is thus to balance the work across the 8 processors without overloading the first-stage processor (which may also receive recycling jobs) while ensuring that the preferred processors are chosen as often as possible. Simply loading a job into the queue with the least workload results in the preferred processor being rarely chosen. A cut-off point was therefore used, such that if workload waiting for a processor was below the cut-off level, the preferred processor was chosen. If the level was exceeded, the workloads waiting at the preferred processor and the alternatives were compared, and the job joined the queue of least workload. Workload was measured by the total processor run time required for the jobs already waiting. This method prevented processors running out of work, while ensuring that the preferred route was chosen when possible.

3.16 Summary

Routing flexibility has been interpreted in two ways:

(a) the provision of alternative machines for one or more operations in a predetermined sequence of operations, or

(b) flexibility in the sequence of operations while maintaining the predetermined machine-operation ordered pairs.

In FMS research, flexibility type (a) is known as routing flexibility and type (b) is known as operation flexibility. FMS is an important case of routing flexibility but varies from typical batch manufacturing by the very low set up times, low work-in-progress, and small batch sizes. Nevertheless, a considerable research effort has tried to establish how to use this flexibility most effectively. Dynamic routing in FMS is almost synonymous with real-time scheduling but very few of the systems proposed employ truly dynamic routing. The underlying purpose of most FMS routing algorithms is to balance workload across the machines but Stecke and Solberg (52) found that some particular characteristics of FMSs indicate that this objective is not necessarily applicable. For example, machines in an FMS have a potentially massive tooling capability, but provision of large sets of tools can be expensive. Hence dividing operations between pools of identically capable machines but maximising the proportion of operations to be carried out at one machine is inherently cost effective and also an effective scheduling mechanism.

However, the most important aspect of alternate routing in FMS is that it exists frequently and is seen to be advantageous. It will be seen in the next chapter on current scheduling practice that alternate routing is generally not considered favourably because of the difficulties in its management.

An area where alternate routes are also very important is packet-switched communications networks. Several rules which have proved successful for choosing between alternate routes in networks can be applied to manufacturing systems.

4. Current Scheduling Practice

Scheduling practice has been evolving in parallel with the development of scheduling theory, rather than integrally. This chapter aims to establish the current role of alternative routes and how they are managed. The author's experience is supplemented by a number of sources (64, 65, 66, 67, 68).

4.1 Material Requirements Planning (MRP)

Early computer aided production management (CAPM) systems in the 1960s addressed the massive volume of data regarding products, assemblies and component parts. Initially the problem was to plan the arrival of purchased raw materials and semi-finished and finished parts to meet the manufacturing programme, and to allow sufficient time for manufacture to meet an assigned due date.

This timing was achieved by working back from the assigned due date, using fixed estimated lead times which would take account of the processing time, setting up time, transport, inspection and queuing times likely to be encountered at each major stage of manufacture. Thus the likely times of requirement of all purchased items could be assessed, and target manufacturing completion dates for each stage could be stated. Calculation took account of desirable batch sizes at each stage, and material was only bought to meet a planned requirement. A major advantage over previous systems was that demand for a component or material could be calculated from all orders where it was used, at the same time.

From a scheduling viewpoint, the shop floor worked to meet the expected arrival time at the next major stage of manufacture, but the order of work processed was left to the discretion of local supervisors or schedulers. Since the MRP system generated a series of orders for either manufactured or purchased items, a job would be considered to be complete on despatch to the customer.

4.2 Manufacturing Resource Planning (MRP2)

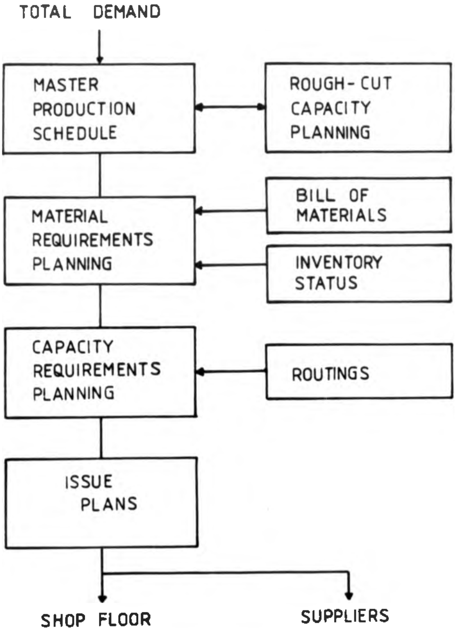
MRP2 systems differed principally from their predecessor systems by their ability to compare the demand on production resources with the availability of those production resources, i.e. capacity planning. Later, "closed loop" systems were developed to feed back shop floor progress into the planning and costing mechanisms.

In contrast to the limited ability of early MRP systems, the capability of MRP2 systems is sometimes considered to have progressed too far towards complete and central control. It is useful to outline briefly the main features of a typical system in order to understand the approach to scheduling and how alternative routes may be handled. A black box diagram of the principal stages is shown in Figure 16.

4.3 Master Production Schedule (MPS)

A master production schedule (MPS) is prepared, comprising all known and expected (forecast) demands for end items from the manufacturing system. These demands include orders to replace stock orders, orders from customers which may be "make to order" or "blanket" orders for regular demands over a period of time, and orders for spares. To prevent unreasonable demands on manufacturing

FIG. 16 BLACK BOX DIAGRAM OF PRINCIPAL ACTIVITIES IN MANUFACTURING RESOURCE PLANNING



resources, and to identify possible future problems which should be addressed now, the manpower and manufacturing capacity is taken into account at the machine group or department level by means of "rough cut capacity planning". Attention must also be paid to any obvious unavailability of material e.g. long leadtime materials or new tooling. Thus only feasible orders should be included for more detailed planning in any period, and estimates of overtime, subcontract volumes, unused capacity or procurement of extra manpower or equipment may be assessed.

4.4 Rough-cut Capacity Planning

Rough-cut capacity planning attempts to ensure the feasibility of the master production schedule. Different techniques are available, of different accuracy, from allocation of some total schedule demand for man and machine hours using a historical percentage split over the resources required, to detailed calculations involving the bill of materials, routing data, time standards of man and machine hours and lead times for all end products and their components.

There is clearly a balance between the accuracy of the calculation results and the effort required to achieve them. The choice of a technique may depend on the complexity of the products, the utilisation of the equipment, and the nature of the market (e.g. predictable demand or highly variable, specialist or highly competitive, etc) which determine the accuracy level which may be necessary or tolerated.

4.5 Materials Planning

Using a similar procedure to that outlined in section 4.1, the MRP module converts the planned volumes of end items by period back into time/quantity requirements for each subassembly and component item using the bills of material and current inventory status, to produce orders for purchased items and made-in items. Common MRP procedures use standard lead times and standard batch sizes at each stage of manufacture, making adjustments for yield as necessary. Thus the output from the MRP system comprises purchase orders, internal works orders, and adjustments to existing orders, stating the quantity and time of requirement. The expected completion time for each manufacturing operation is available for detailed planning.

4.6 Capacity Requirements Planning (CRP)

The objective of detailed capacity planning is to "reconcile" demand for manpower and manufacturing capacity from the MRP plan with the availability of these resources in a period. In practice, actual capacity is rarely fixed, and is subject to fluctuations on a daily basis. A key worker may be absent. Machines break down. The efficiency of production can vary. A key tool may break. Capacity may be increased by:

- overtime
- redeployment of manpower between work centres
- bringing on extra shifts/crews
- subcontracting

* buying extra equipment

(Capacity may be decreased by the opposite of these activities where appropriate).

Although gross mismatches should have been avoided by rough-cut capacity planning on the master production schedule, some compromises and approximations at that stage may manifest themselves as over or under demand at the detailed level.

In addition to the data used by the more comprehensive rough-cut capacity planning techniques, i.e. bill of materials (BOM), routing, time standards and lead times, the following data is used for capacity requirements planning and is largely supplied from the MRP module:

* batch sizes
* current status of all shop orders in progress (WIP level at each stage of manufacture)
* lead times of shop orders in progress and planned future releases
* expected completion times of each operation of each current shop order (and planned releases)

The routing file or operations file supplies data on the sequence of operations, the work centre location of each operation, and also the set up time, run time, average queuing time and transport time. The work centre file supplies data on the availability of each machine or work centre and alternative work centres.

Reconciliation may involve increasing the capacity available for which some options were listed above (or decreasing the capacity by doing the opposite where possible) or may involve changing the demand on that capacity, by one of the following actions:

* pulling operations into an earlier time period (and assessing the feasibility of preceding operations)
* pushing operations into a later time period (assessing effect on the due date)
* planning operations onto an alternative work centre (if available, and if there is available capacity)
* changing the due dates promised to customers
* subcontract job/operations
* split batches to produce only urgent quantities required in that period
* review make or buy decisions

Therefore, in computer aided production management, the use of alternative routes first appears in the reconciliation stage of capacity requirements planning.

Two broad philosophies exist to manage this reconciliation:

* infinite capacity planning, where all due dates are considered to be fixed and must be accommodated until the loading is recognised as impossible to achieve
* finite capacity planning, where the limits of capacity are taken into account when planning which jobs are to be carried out in which period.

Before describing these activities, definitions of capacity planning and scheduling will be clarified.

Capacity planning considers whether all the tasks required in a period can be made in that period or not. The priority of any order, either in preference to another order, or with regard to its operational sequence is not considered beyond the due date limits supplied by the MRP system which enabled CRP to fit the order/task/operation into the right time slot.

Scheduling determines the sequence in which tasks should be carried out for all those tasks required in a period, and takes into account the structural and sequential dependencies of those tasks.

4.7 Infinite Capacity Planning and Backward Scheduling

Using the output from the MRP module, operations are loaded into work centres, working backwards from the due date, and taking into account processing times, queuing times and transport times. If the total time required to complete an order exceeds the time available between current date and due date, the procedure will be repeated using "compressed" queue time to take account of the effect of expediting urgent orders between work centres. Clearly, the proportion of planned "expedited" orders must be carefully monitored.

Under infinite capacity planning, no account is taken of the capacity constraints. Adjustments may have to be made to the capacity, using the methods listed above, in order to meet the demand. If the capacity cannot be adjusted adequately, then the demand on that capacity must be adjusted by the other methods listed. A close eye must be kept on the reality of the plans and any accumulating backlog.

4.8 Finite capacity planning and forward scheduling

Finite capacity planning takes account of the limited resources available in each time period. It is usually associated with forward scheduling.

Working forward from the current time period, any overdue orders are loaded into the plan first. Further orders are loaded into the time periods according to some priority rule. Clearly, finite loading takes account of the capacity available but, in a similar manner to infinite capacity planning, the level of any accumulating backlog must be monitored carefully in order to determine whether some capacity adjustment is required. Capacity may also have to be adjusted to modify unacceptable due date predictions.

4.9 Work Centre Scheduling

A job is planned into a work centre or onto an alternative work centre at the capacity planning stage to be made in a particular time period. The actual allocation of a job to a particular machine is sometimes called "job loading". Work centre scheduling is carried out on a short-term basis to produce work-to lists which show the exact sequence of jobs and operations on each machine and which are issued to shop supervisors. Work centre scheduling examines the list of operations required at a work centre over the next shift or day or few days. These operations may be for jobs already in progress or for new jobs which have just been released for manufacture. The list of jobs is ordered by priority, typically using one or more priority or sequencing rules. Kochhar (68) stated that this should be carried out on a shiftly basis. Vollman et al (64) consider that rescheduling each shift is impractical from a computer operations viewpoint, and suggest 2-3 times per week.

The "work-to" list identifies the jobs required and their operational due dates. The operational due dates may have been revised from the original plan during the most recent running of the MRP module. In contrast to the routing data which is considered to be static and is issued on the shop paperwork for the job, due date information is accepted as being dynamic and is not printed on the shop paperwork.

4.10 Cellular Manufacture

In the last ten to fifteen years, a number of companies have moved from a process layout to a product layout in the medium variety, medium volume section of the product-quantity curve in order to achieve for batch production some of the advantages traditionally associated with line production e.g. dedicated facilities, short path length, small transfer batch size, and very importantly, complete manufacture of the component by one work group which encourages feelings of responsibility for quality and ownership of any manufacturing problems. Routings have therefore been dramatically simplified.

By dividing the range of components produced in a factory into families of parts of similar design or which follow similar production routes, the group of machines required to produce that family of parts can be isolated in a cell. Major benefits are claimed including reduced set up times, shorter lead times and hence lower work-in-progress (and usually lower machine utilisation too) and easier and more visible scheduling. Independence and self-contained solutions are encouraged, in order to promote the business identity of the cell. After receiving MRP output at the factory level, where some capacity planning has been carried out, the cells must make their own detailed assessment of capacity and create their own schedule to meet this top level demand. The objective is usually to operate a simple but effective system. In these operations, it has been found that alternative routes are discouraged because they increase complexity (of scheduling and tracking), and compromise quality - one machine will always be better (69).

4.11 Computer Aided Process Planning (CAPP)

Aside from a few research installations, there are few working examples of computer integrated manufacture yet. However, many companies have installed some of its component parts but without full linkage between them. One of those components is computer aided process planning which determines the best manufacturing route for a part design. Generative process planning, which will automatically determine the best manufacturing method from the design coordinates of a 3D CAD model and a manufacturing database is still at the research stage. However, the computer assisted variant approach (70), which enables recall of process plans for similar parts, carries databases of manual, historical and machine data and which carries out calculations quickly, is in common use.

Under this method, one or more operations is assigned to a machine type with a known tool requirement and with known time requirements for setting and operating. The operations required to produce an assembly or a part are gradually built up and stored as a list which includes the machine and tooling requirements. This list can then be downloaded into the MRP2 system, specifically into the routing file. In general, alternative operations are not allowed because of the difficulties of file maintenance. In addition, it may be argued that the CAPP software identifies the optimum manufacturing route and hence the optimum machine for each operation, for which data on costing and job timing are quickly accessible (71).

4.12 Optimised Production Technology (OPT)

OPT is a proprietary software package for manufacturing planning and control. It is capable of many activities similar to MRP packages, but differs from MRP primarily by its focus on "bottleneck" operations, and a different control philosophy (73).

Using the BOM and routing data, a module called "Buildnet" constructs a network which attaches operations data to each part in the product structure (64, 74). Alternative routes are allowed (1, 72).

This OPT network is then combined with the MPS and a form of rough cut capacity planning is carried out. The average expected load on each resource is calculated as the average capacity requirement by period over the planning horizon, divided by the average capacity available, and is therefore some sort of ratio (64).

The resources are then sorted in order of average expected load. The purpose of this stage is to identify the most heavily loaded resources, i.e. the bottlenecks. The bottlenecks are scrutinised against the following criteria(64):

* is the data correct?
* are the time standards accurate?
* can the capacity be increased?
* can alternative routes be used for some items?

This capacity planning stage is re-run until no further changes have been made, at which point the "true" bottlenecks are identified (64).

The network is now split into two parts (64, 73, 74):

(i) A "critical" resource portion which comprises bottlenecks and all succeeding operations. These operations will be forward scheduled from the bottlenecks under finite loading constraints.

(ii) A "non-critical" resource portion comprising all other operations which will be backward scheduled from the bottlenecks using a type of MRP logic.

The above procedure outlines the mechanics of the software only. Successful implementation of OPT demands adherence to a set of rules which present an alternative perspective on material flow in manufacturing (72). For example, "an hour lost at a bottleneck is an hour lost for the whole system". Improving flow of material through a bottleneck will allow more material to pass through operations before and after it. This clearly indicates that alternative routes, which either eliminate or alleviate the bottleneck, have a critical role to play.

OPT allows pools of identical machines, and also allows "machine substitutes". The substitutes are linked to a main machine and have some efficiency ratio compared to the main machine which is based on a ratio of their output rates (I). The pools of machines are treated as options of equal weighting. The machine substitutes are always linked to a main machine, which may be considered to be the preferred machine, among single operation alternatives.

Data on alternative routes, pooled machines, or substitutes particularly, is only required for bottleneck machine resources. The capacity planning iterations ensure that alternative routes are activated as required. It is likely that after activation of the substitute for one machine and re-running the capacity planning software, another resource would become the bottleneck. This machine may not have any substitutes and will dictate the flow of work through other machines before and after it, including the previously critical resource and its substitute(s).

OPT handles alternate operations at the capacity planning stage, and then schedules jobs onto the alternate operations, either forwards or backwards according to whether those resources remain as bottlenecks or not. A work-to list is usually issued at the start of each day or shift in OPT environments listing which jobs to load onto which machines and in which order. Any use of alternate routes is therefore fixed on preparation of the work-to list, but not at job launch, allowing assignment of a job to a machine during the next few days, rather than several weeks hence. It might be argued that breakdowns are unpredictable and that local reassignment of a job to a substitute may be necessary during a shift to ensure completion of the work-to list, if not compliance with it.

The OPT mode of operation is not quite Russo's "level 2" point of transition between operations but is superior to "level 1" decisions before launch.

4.13 Summary of the Role of Alternative Routes

Alternative operations or machines were seen to have an important role during capacity planning. Many CAPM systems do not allow alternative routing because the design and maintenance of alternative routing files can be onerous (64). An example of the difficulty caused by the lack of alternative routing facilities is illustrated by the experience of Westland Helicopters in the early 1980s. One of the primary reasons that the company was unable to find a CAPM package that suited their operation was the inability of commercial packages to cope with alternative machines (75). The company felt compelled to write its own capacity planning system. This system directed jobs which had many alternative operation locations away from machines which had few alternatives, in order to maximise machine utilisation. Since writing and installing this program, successive reviews of Westland production facilities have now resulted in the introduction of cells and consequent removal of alternative routes in order to reduce complexity (76).

During rough cut planning, it is likely that the alternative routes can be aggregated together into the same work centre in order to plan work into them as a unit. Difficulties may arise if the machines in that group have different capabilities, which is often the case with alternative machines.

During capacity requirements planning, the task becomes a "loading" problem in terms of which machine to choose for a particular job. Scheduling then prioritises and orders all the jobs allocated to a particular machine. Any alternative routing is thus now fixed into the schedule and is issued on the job paperwork as the list of operations.

It is considered that the route should be able to be revised in exactly the same way as the due dates are revised. Routing data is dynamic and should not be treated as static data. As less paperwork is issued for a job, the only important data item is a job's identity.

All information about the job including route, operations, tools and due dates can be downloaded from any local terminal on entry of the job identity. One of the important current objectives when reviewing manufacturing systems is to maximise "flexibility". Flexibility can be manifest in many ways, of which one is alternative routing. Carter (77) examines different types of flexibility by classifying them into "timeframes" in which they have an impact on production (very short, short, medium and long term) and "incentives" (insurance, economics and strategy). He views routing and operation flexibility as a very short term, insurance measure, where "insurance" means protection against uncontrollable variables such as breakdowns and uncertainty of demand. In addition to the usual problems of breakdowns, discovery of defective material and late receipt of material or tools, a case for using alternative routing as a protection against downstream part shortages is made, by using alternative machines to maintain material flow. Routing flexibility is also viewed as a means of achieving mix flexibility if mix changes alter workloads and cause a bottleneck which can be relieved by an alternative route.

Thus, alternative routes are used at three levels in practice:

(i) Capacity planning:

to alleviate or eliminate a potential bottleneck

to adjust capacity

to meet volume changes

(ii) Operations scheduling

to meet the due date plan

to make the work-to list and allocate a job to a machine or work centre

(iii) Shop floor control

to counter the effects of breakdowns, sudden demands and other unforeseen problems.

5 Development of Alternative Routing Strategies

In this section, a number of heuristics will be developed for testing. In addition, their expected performance in different machine configurations and expected effects will be discussed.

5.1 Summary of factors to be considered

From previous sections, a number of characteristics must be considered when addressing alternative routing decisions:

5.1.1. Routes may be of equal desirability (equal weighting) or one route may be preferred.

5.1.2 An alternative route may comprise a single operation or several operations.

5.1.3 Buffers serving the alternative routes may be pooled or dispersed.

5.1.4 Alternative routes may be composed of identical or non-identical machines.

5.1.5 The routing strategy should have an effect which is in line with the production objectives. Different strategies may be effective towards different objectives.

The objectives of a 'good' routing strategy are identical to the general objectives of scheduling which are considered to be:

5.1.6 Minimise overall flowtime and prevent 'tails' (very early or very late jobs).

5.1.7 Minimise overall machine utilisation (by using the fastest machine where possible).

5.1.8 Minimise the number of set up changeovers and hence the total setting time.

5.1.9 Minimise the effect of delays and disturbances e.g. effect of breakdowns or setting up.

5.1.10 Minimise the queuing time.

5.1.11 Minimise the inventory

5.1.12 Minimise lateness.

5.1.13 Minimise machine idle time.

In addition, some characteristics of a good routing strategy may be defined:

5.1.14 Easy to calculate

5.1.15 Minimise the amount of information required for a good decision.

5.1.16 Maximise the chance of making a good decision based on the information available.

5.2 Equally weighted options

Equally weighted options imply that no one alternative route is favoured. It does not necessarily follow that a job should have an equal chance of being directed to each alternative. This equal chance is only true for routes of identical machines and operational capability. In practice this case rarely occurs, e.g. due to different breakdown histories or availability of tooling.

Hahne (47) has shown that 'workload balance' was important and indeed, it seems intuitively obvious that the objective of the equal weighting alternate routing decision should be to balance the workload across each partial route. However, the method of achievement of workload balance is not so obvious, and a number of methods are proposed for testing.

5.3 Single operation alternatives

Considering the case of single operation alternatives first, the simplest configuration is composed of identical machines drawing work from a common (pooled) buffer. This is analogous to the FMS loading problem described by Stecke (52) where the 'first free' machine rule was determined to be the most effective. Within the pool or buffer, jobs are ordered according to the priority rule in operation. As a machine becomes available, the next most important job is loaded from the buffer. This seems to be an obvious mode of operation and would also balance work across non-identical machines (since faster machines would draw new jobs more frequently). The main disadvantage of this method is that no attention is paid to minimising the number of set up changes. A common feature of FMSs is that set up penalties are low and consecutive batches of dissimilar requirements can be easily tolerated. The level of setting up time that can be tolerated where setting up time is not minimal depends on the capacity available. The level of utilisation of that capacity will have a direct effect on inventory, queue time and flow time. Hence a variant of the "first free" machine rule will be used where a machine which has just completed a job will scan the queue in priority order for the next job, but a job which arrives when more than one machine in the group is idle, will attempt to find a machine which has the correct setup already. If a machine with the correct setup is not found, a set up change will be initiated. This reasoning would be followed in practice.

Following the workload balance concept, an assessment of the workload represented by the queue at each alternate machine could be made, or an assessment of the expected completion time could be made which would take into account the progress of any current job on the machine. The current job could have been started only recently or may be about to be completed. Progress of the current job could be unimportant in the long term. Similarly the exact workload in the queue might be an unnecessary effort of calculation and merely the number of waiting jobs might be adequate. Three possible decision rules are derived:

5.3.1 Join the queue with the least number of jobs

5.3.2 Join the queue with the least anticipated workload

5.3.3 Join the queue which will lead to the earliest expected completion of the job.

Hahne (47) discounted any alternative route which was currently experiencing a machine breakdown. This is not considered to be useful. In practice, an assessment of the severity of a breakdown can usually be made within 15-20 minutes of the arrival of the repair crew. After this period, it is known that the machine has already been restarted, or is likely to be down for a short or long time. Thus the expected duration of the breakdown should be considered, in order to direct more work to the delayed processor near the end of the stoppage.

In a similar manner to the effect of setting up times on the efficiency of the 'first free machine' rule, each of these three rules could cause unnecessary set up changes. For this reason, a rule was devised for a press shop (78) where a job is directed to the route that contains most work-in-progress of this job type. If similar jobs had not already been directed to one partial route, then some other rule is required, e.g. choose the route with the least work-in-progress, or least machine load.

An alternative approach to single operation options could consider the processing capability of non-identical machines. For example, if one machine worked twice as fast as the second machine of two options, twice as many jobs should be directed to the fast machine. Some predetermined ratio could be used to direct jobs between options, based on some generalised assessment of their capability. Workload balance may not be exact because of variety in product mix and varying abilities for different operations, and this faster crude ratio decision might be adequate. A ratio may not address setting up times but it would ensure a steady supply of work to each alternate. The ratio rule is analogous to the split traffic routing algorithm in communications networks (56). Where the ratio allows each route an equal chance, it is equivalent to the "round robin" rule for data communications networks tested by Ephremides et al (57).

5.4 Partial route alternatives

Continuing the discussion of routing strategies for equally weighted options, the derivation of rules for single operation alternatives can be extended to options comprising one or more operations.

Clearly, the 'first free machine' rule should not be used for partial routes of more than one operation. A case was observed (63) where the first operation on the partial route was very fast and supplied a slow operation which was slower than the single operation alternative. Too much WIP is drawn by the first operation into the route which is slower overall. Even if the partial routes are identical, a decision should take into account the capability of the constraining or 'bottleneck' machine on the partial route.

During capacity planning, a 'bottleneck' machine is identified by a heavy planned workload, and during operation, a bottleneck machine is characterised by high utilisation and a long queue of work. A workload balance approach must therefore address the bottleneck machine in any partial route to prevent overload.

Extending the decision rules proposed for single operation alternatives, the following decision methods can be formulated:

5.4.1 Select the route with the least number of jobs yet to be processed through the bottleneck machine.

5.4.2 Select the route with the least anticipated workload for the bottleneck machine.

5.4.3 Select the route which is expected to lead to the earliest completion of this job through the bottleneck.

Other methods proposed for the dispersed buffer single operation options may also be similarly extended:

5.4.4 Direct jobs to alternates according to some ratio of processing capability of the bottleneck machines.

5.4.5. Direct jobs to the route which contains the greatest work-in-progress of this job type. If no work in progress exists, use another rule eg. least workload for the bottleneck machine.

Reviewing the adaptive rules in section 3.10 for packet-switched networks, the "isolated shortest queue" rule is equivalent to choosing a route which contains the least number of jobs. The other adaptive rule "backward learning technique" could be implemented by feeding back the flowtime such that a job is allocated to the route displaying the least flowtime. A further rule is formulated:

5.4.6 Calculate the average flowtimes achieved on each route and direct the job to the route which currently achieves the least average flowtime.

In all cases of alternative routes where at least one of the alternates has more than one operation, only dispersed buffers will be tested. In practice, jobs would be allocated to one partial route, representing dispersed buffers.

5.5. Preferred routes

If one of the alternatives is a preferred route, the workload balance concept is replaced by an objective to send as much work as possible to the preferred route without overloading it.

Further extending some of the previously proposed strategies, the following decision methods have been formulated:

5.5.1 Use some ratio to force a proportion of the jobs, through the preferred route. The ratio should take account of the capability of the bottleneck machine compared to the total demand.

5.5.2 Direct all jobs to the preferred machine until the workload in the queue exceeds some level, which triggers diversion of the next job to the alternate. This "cut-off-point" could be related to the processing capability of the alternate route, or some arbitrary assessment of unacceptable queuing time.

5.5.3 Send the job to the route with the greatest work in progress of this job type (to minimise set up changes). If there is none, choose the preferred route. This rule could be used to advantage with a work mix of several job types by indicating different preferred routes for different job types while still allowing alternatives to be selected when necessary. Although it has been assumed so far that a preferred route is inherently advantageous because of overriding factors such as speed or quality, it would be possible to nominate more critical job types (in terms of quality or schedule achievement for example) to the preferred route and less critical job types could actually be nominated to less advantageous machines as their own "preferred route."

5.6 Random assignment

Assignment of a job to an alternative route at random will be used as a comparator against which the usefulness of all other rules can be measured.

Random assignment clearly takes no account of the state of the network at the time of the decision. By its nature, routes are considered to be of equal weighting and although it does not address preferred routes, it could be used to check that preferred route strategies really are favouring the preferred routes.

The route is decided by sampling from a uniform distribution.

6. Description of the model used for experimentation

Many different approaches to examining scheduling problems were reviewed in chapter 2. It was observed that most of these approaches made simplifying assumptions in order to achieve a solution and that scheduling problems are inherently difficult. The additional complexity of alternate routes forces attention to simulation as a tool for investigation. A number of different computer simulation packages are available at Warwick University, from which a package capable of accepting potentially complex decision logic was chosen.

The model is written using See-Why, a general purpose simulation modelling package marketed by Istel. See-Why is a collection of subroutines which can be linked together by user-written Fortran code (80). It is a visual, interactive modelling package allowing the user to watch the progress of each simulated event on the computer screen and to interrogate system status at any time. These facilities are particularly useful for debugging and understanding the interactions occurring within the model.

The experimental model was based on a generic model constructed for a machine shop (79). A complete listing is attached as appendix 1. Some attempts were made to use data from industrial sources, but alternative routes are often used informally and because of the difficulty of maintaining routing files, operational data is not available.

The basic functions and data structure of the new model are broadly similar to the base model but the control rules and performance recorders have been largely changed to meet the requirements of this work.

Primary functions of the simulation model may be classified as:

(i) modelling of physical processes - machine and material

(ii) control of material between operations

In addition, a simulation model has secondary functions such as:

(i) display control

(ii) handling of input (data) files

(iii) interactions to investigate performance or to change parameters

(iv) collection of performance data

(v) handling of output (performance) files

The "physical process" functions are mostly independent of the control routines in their operation and will be described first.

6.1 Moving jobs between machines

On arrival or completion of processing, a destination is determined, which may be "machine for the next operation" or "exit the system". The method of determining the destination is a control routine. The "move" function moves the job from its previous location to its next location, if required. During transit, the job may not be diverted elsewhere. The travel time to the next operation is supplied in the operation data file.

6.2 Processing a job on a machine

In order to start processing, the following conditions must be satisfied:

(i) the machine must be idle (no other state is acceptable)

(ii) the machine must have the correct set up for the job

(iii) material must be available at the machine.

During processing, a machine may be subject to a breakdown or a tool change. A breakdown or a tool change will cause a delay to processing. It is assumed that no parts are scrapped or damaged as a result of the delay. Once started, a pallet of work will not be interrupted due to the arrival of a more urgent job (no pre-emption). The processing time is calculated as the pallet capacity multiplied by the operation time per component. Both these data items are supplied in the operations data file. Operators are assumed to be of uniform ability.

6.3 Setting up

Changing the set up on a machine requires a certain time, which may be specified for each set up, but which is constant regardless of the previous set. Therefore, no account is taken of any sequential set up dependency. It is assumed that set up changes may be started as soon as they are required which, in some cases, where setting skills are required, assumes unlimited availability of machine setters. Set up changes are instigated by the arrival of a job at a machine which does not presently have the appropriate set up, or by the choice of a job, from a queue of waiting jobs, for which the current set up is not appropriate. There is no "look ahead" facility to anticipate what set up changes should be started to prepare for jobs which are expected to arrive shortly. A set up will not be started if another machine is already processing a job on this set up and is expected to finish this job before another machine could be set up. Machines which are broken down are not included in the assessment.

6.4 Breakdowns

Breakdowns occur according to the "mean time between failures" data for each machine, and the distribution of repair times for that machine. Time between breakdowns is sampled from a negative exponential distribution. At the time of the breakdown, the repair time is sampled from an Erlang distribution. It is assumed that any part in the machine at the time of the breakdown is not damaged or scrapped. Breakdowns can only occur when the machine is working. "Time to next failure" recording considers only working time. Processing of a pallet being worked at the time of the breakdown continues after the breakdown without penalty to the remaining processing time. The number of fitters is assumed to be unlimited, or conversely, any waiting time is already included in the repair time data.

The breakdown level is given by the total breakdown time in a period divided by the sum of the breakdown time and the total working time. A breakdown level of approximately 10% is used here.

6.5 Control routines

The control routines are clearly the most critical and determine exactly how material flows through the shop i.e. route and rate. The principal decisions which determine flow are:

(i) choosing the next job for a machine - "what next?", this involves assessment of job priority

(ii) choosing the next machine for a job - "where next?", this involves choice between alternative routes

Secondary control routines affecting these decisions are:

(i) determination of travel times

(ii) deciding when to change a machine set up

(iii) deciding to which machine of a group to allocate a job.

6.6 Choosing the next job for an idle machine

It has been stated earlier (chapter 2) that the "what next?" decision has been studied extensively and the objective of research in this area has been to identify the most effective method (according to some criteria) of assigning a priority rating to each job in the queue for this machine. Priority ratings become particularly significant when machine utilisation is high (and consequently work-in-progress levels and queuing times are high).

Dispatching rules have been discussed earlier and are essentially a means of ordering jobs in importance, according to some criteria, which are usually due-date or flow time driven. At the time of each selection of the next job, the queue is reordered according to the dispatching rule in operation.

A small number of basic rules are available in this model:

(i) FCFS (first come, first served). Where possible, jobs are treated in the order in which they arrive.

(ii) SIO (shortest imminent operation). Jobs are ordered according to the expected total processing time for the next operation, least first.

(iii) Achievement ratio. The sum of the number of completed components plus downstream work in progress is divided by the planned requirement to give a measure of achievement. Least achieved jobs are most important. This rule has been used in a number of industrial models written by Warwick University Manufacturing Simulation Group where repeat batches are processed in turn through the same facilities to meet some daily or weekly requirement.

(iv) Urgency ratio. The achievement ratio is divided by the remaining operational lead time to give a measure of ease of achievement. Low resultant values (from low achievement, or long lead times) indicate a job of greater importance.

(v) Maximum remaining operations. Jobs with a high number of remaining operations are given priority. The assessment of remaining number of operations must take any partial routes into account. If the job is already on a partial route, only operations remaining on that route plus any downstream operations will be considered. If downstream operations contain alternate routes, only the first route will be considered plus any previous or subsequent operations to that partial route. The data file is ordered such that any preferred route is listed first.

(vi) Launch time. Due dates are not assigned to jobs in this model. Assuming that jobs are launched at some expected lead time before they are required, it is taken that jobs launched early are more important than jobs launched later. The job launched earliest is given highest priority.

The sequence of actions for choosing the next job for a machine which is idle is as follows:

> If this is not a dedicated machine, and any stated minimum quantity of components has been achieved, it is possible to change the set up if required, So:
>
> Search the queue in order of priority according to the dispatching rule being used starting with the most important job. Find the first job which requires an operation of which this machine is capable. If the machine already has the correct set up, processing can start. If not, then a set up change is attempted. If the criteria for allowing a set up change are not met, the next job in the queue is examined.
>
> If no material is found for any of these operations, the machine remains idle.

To prevent continual set up changes with the arrival of each pallet, a minimum batch quantity can be assigned to operations as required. This prevents a set up change until the specified minimum quantity of components has been processed.

The quantity may be equal to, or some multiple of, the pallet capacity. This minimum quantity is clearly inapplicable to special purpose or other dedicated machines.

6.7 Choosing the next operation for a job

On arrival in the shop, or on completion of an operation, the next operation for a job must be determined. Parts of the route, or indeed, the whole route, may be fixed. One or more operations, possibly sequential, may be required to be carried out on the same machine (but with different set ups).

For an operation which has no alternatives, the following options must be considered:

(i) operation is carried out on only one possible machine

(ii) operation is carried out on one of a group of machines dedicated to this operation

(iii) operation is carried out on a machine which also carries out other operations

(iv) operation is just one of a collection of operations assigned to a group of machines. One or more of the machines may be capable of one or more of the operations.

The general case described in (iv) is followed in this model, since it naturally handles (i), (ii) and (iii) anyway. The procedure is as follows:

Identify the next operation and any alternative operations.

Choose from among the alternatives according to the rule being used.

Identify the machine group where this operation takes place.

Order the queue (if any) waiting for machines in this group.

Taking the members of the ordered queue in turn identify all the machines in the group which are capable of this operation and which are idle.

If a machine is dedicated to this operation, or the minimum number of batches/components has not been achieved, start this machine - priority 1

If a priority 1 machine cannot be found, is there a machine with the correct set up already? (i.e. minimum has been achieved) - priority 2

If a priority 2 machine cannot be found, attempt to start to change the set up - priority 3.

6.8 Inter-operational transport time

A fixed travel time is used to move jobs between operation locations, and is the same for all movements in any one experiment. Different magnitudes of travel times could be tested to investigate the influence of travel time on the quality of the decisions made by the alternative routing strategies. Different travel times can be tested for each inter-operational movement.

6.9 General demand and control philosophy

The two principal ways of managing the arrival of new jobs into the system are to use some sampling process to generate inter-arrival times at a rate of arrival less than the overall rate of processing, or to allow an unlimited store of potential entrants from which the next job is drawn as required. Variants of these two extremes do exist and it is important to note that the second policy is only stable if finite buffers are used (or more work will be drawn into the system than can be processed).

The first policy aims to maintain some predetermined utilisation level, while the second policy aims to find out what is the maximum capacity or output level. Flowtime can be investigated in both cases, but will be highly inflated under the second policy because of the high utilisation and work-in-progress levels, unless buffer sizes are carefully managed and limited.

The first policy has been adopted for this model, in order to achieve a realistic machine utilisation level.

6.10 Data Files

Data for each experiment is entered through initialisation, machine and operation files. Examples of these files are contained in appendices 2, 3 and 4.

The initialisation file contains data items required to set up the See-Why array, and also contains the names of the machine and operations files.

The machine file includes the following data for each machine:

Code number of cell or group in which it is located

Number of machines in the group

X and Y coordinates for the screen display

Mean time between failures

Mean time to repair

The following data is supplied for each operation of which the machine is capable:

Set up code

Minimum number of batches which must be machined on this set up before a set up change is allowed

Set up time

The operation file includes the following data for each component:

Descriptor

Component number

Starting cell or group number

Periodic demand (in components, not batches)

Pallet capacity or minimum process batch size

Screen display colour code number

Inter-arrival time

Random stream number for arrival time sampling

The following data is included for each operation required by the component:

Workstation location

Set up code

Operation time

Travel time to next operation

If this is an alternative operation, how many lines of the routing should be skipped to find the next operation

If this is an alternative operation, what is the set up code of the first operation on the first branch (i.e. to locate the beginning of the alternatives on the operations list)

If this is an alternative operation, what proportion of the total number of batches should be sent on this route (for use by the ratio rule)

Up to 3 alternative operations, specified by their set up codes, are currently allowed.

6.11 Interactions

At any point during execution of the program, execution may be temporarily halted in order to "interact" with the model. A number of interactions are available in the See Why software which allow the user to investigate the status and location of model entities, e.g. machines and jobs. In addition, the user may add Fortran code to create further interactions, which could be used for example to change parameters or display results. The following user interactions were written:

6.11.1 "PRAM" to examine the main parameters of the current experiment. A typical screen display is shown in figure 17.

6.11.2 "CDSP" to change the part of the shop being displayed if more than one screen is required to display all the machines

6.11.3 "MCHQ" to query the proportion of time spent in each state by each machine since the start of the current period. A typical screen display is shown in figure 18.

6.11.4 "PRDQ" to query the level of production so far. A typical screen display is shown in figure 19.

6.11.5 "LDTM" to examine the makeup of lead times or flowtimes observed so far. A typical screen display is shown in figure 20.

6.11.6 "REPT" to obtain a summary report of data collected so far. A typical summary report is attached as appendix 5.

6.12 Performance measures

A variety of performance measures are used in this model, which attempt to measure the degree of success of different alternate routing rules against the range of manufacturing control objectives described earlier.

6.12.1 Machine activity

At any time, a machine can be in one of 5 possible states:

1. Idle - no material

2. Working

Figure 17 Typical screen display for "PRAM" interaction

```
   Present parameters:

 1   Dispatching rule    (1-4)      =        1
 2   Alternate route rule (1-6)     =        1
 3   Replication number             =        1
 4   Travel time method             =        0
 5   No. warm up periods            =        5
 6   No. periods in experiment      =       50
 7   End of simulation              =   550000.0
 8   Period duration                =    10000.0
 9   Current travel time            =       30.0
10   Current set up time            =       30.0
11   Review interarrival times?

Select line (0 to accept)
```

Figure 18 Typical screen display for "MCHO" interaction

Time in this period = 200000.0

M/C#	%IDLE	%WORK	%SETU	%DOWN
1	14.2	64.9	13.1	7.8
2	42.9	41.7	10.8	4.6
3	42.5	41.6	10.5	5.4
4	13.9	64.7	13.0	8.3

Key:

M/C# = machine number

Percentage of time so far spent in these states:

%IDLE idle, i.e. starved of work

%WORK working

%SETU being set up

%DOWN broken down

Figure 19. Typical screen display for "PROG" interaction

Time in this period = 200000.0

			WIP		
COMP	FLOWT	OUTPT	NOW	MIN	MAX
1	1236.0	685	1	0	18
2	1368.8	436	0	0	11
3	1549.4	364	1	0	13
4	1538.4	270	0	0	7

Key:

COMP	component
FLOWT	flowtime
OUTPT	output, number of batches completed so far
WIP	work in progress
NOW	number of batches currently in the system
MIN	minimum number of batches observed in the system since start of current run time
MAX	maximum number of batches observed in the system since start of current time

Figure 20 Typical screen display for "LDTM" interaction

Time in this period = 200000 0

CP	%QUEU	%WORK	%BKDN	%TLCH	%TRAV	NOWIP	NOFIN
1	80.2	13.3	1.7	0 0	4.9	1	685
2	75.8	17.6	2.2	0.0	4.4	0	436
3	75.5	18.4	2.3	0.0	3.9	1	364
4	67.9	25.3	2.9	0.0	3.9	0	270

where:

CP component number

Percentage of time spent in the following states:

%QUEU queuing

%WORK being processed

%BKDN processing interrupted due to a breakdown

%TLCH processing interrupted due to a tool change

%TRAV travelling

NOWIP number of batches currently in progress

NOFIN number of batches completed so far

3. Being set up

4. Broken down (being repaired)

5. Blocked

Time recorders exist for each machine for each state. When a state change occurs, e.g. start of breakdown, the elapsed time spent in the previous state is added into the time recorder and the time of the current change of state is stored.

At the end of the period, the elapsed time spent in the final state is added into the recorder for that period. At the end of the experiment, the accumulated time in each state is divided by the length of the time period to yield the proportion of time in each state in each period.

It should be noted that the "idle" state can only arise through shortage of material.

6.12.2 Set up change recording

For each set up of which a machine is capable, the total number of times that the set up is carried out and the total number of pallets machined on that set up in each period is recorded. These results show which job operations are causing the set ups and give insight into the dispersal of job types between routes or groups of machines.

6.12.3 Flowtime

The launch time of each pallet is recorded. On completion of all operations, the elapsed time since the point of launch is recorded. A flag is attached to the pallet indicating which route was taken. Thus flowtimes can be collected by route and job type. In addition, the maximum and minimum flowtimes for each job type in any period are recorded to note the spread of flowtimes caused by disturbances such as breakdowns.

6.12.4 Flowtime analysis

In a similar manner to machine state recording, jobs were also allocated states, and the time spent in each state was recorded:

1. Queuing
2. Being worked
3. Waiting during a breakdown
4. Travelling to the next operation

Total travel time depends on the number of operations and the inter-operational transport time in use. Waiting during breakdowns depends on the breakdown pattern being used. Queuing time will depend on the overall level of work in progress. In general, the proportion of time being worked should be maximised as a proportion of flowtime but when choosing alternative routes, a slower option may be chosen to reduce queuing time and expected overall flowtime.

6.12.5 Work in progress level

The average level of work in progress for each job type in each period is calculated by accumulating products of a level of work in progress (WIP) and the time elapsed while that level of WIP was current. At the end of the period, the grand total of the products is divided by the period time to yield the average WIP level. At the end of the period, the residual WIP level at each machine is recorded to make any blockages visible.

6.12.6 Output recorders

The number of batches of each job type finished in the period is recorded together with a subtotal of number of jobs completed by route.

7. Validation of the simulation model

Before experimentation can begin, the model must be "validated". Carson (81) distinguishes between verification, validation and credibility:

(i) Verification is the responsibility of the modeller to ensure that the model performs in the way in which the modeller intended it to perform

(ii) Validation is the responsibility of the user (domain expert) and the modeller to ensure that the model is a representation of the domain with sufficient accuracy to be able to test different conditions

(iii) A "credible" model is accepted by the domain expert as being valid, and can be used as an aid in decision-making.

It appears that Carson's definition of credibility involves a higher level of confidence and user acceptance than is generated by validation. Law and Kelton (82) similarly offer three stages of examination, starting with verification and validation as above, but having a third stage called "output analysis". Output analysis is concerned with determining a simulation model's true parameters or characteristics, which may not necessarily be the characteristics of the system. Output analysis should therefore be continued until acceptable confidence limits can be obtained for the results such that they are "credible".

In this work, the author is the modeller and the domain expert. Since the data is fictitious, it is difficult to rigorously prove validity and credibility. However, statistical analysis of the output yields a good insight into the most important factors in the decision-making process for alternative routes and identifies those factors to which the results are most sensitive.

7.1 Verification

Verification generally refers to the process of testing and checking the computer code to "verify" that it truly represents the assumptions and data accurately (81, 82). Standard debugging techniques are employed, and Carson also suggests:

(a) Structured programming techniques

(b) Program testing under a wide variety of input parameters, to check for extreme reactions and special cases

(c) Collection and display of statistics, over and above those of interest for experiment, in order to expose any possible sources of error

All the above techniques were used for this program. The following points from Law and Kelton enhance Carson's list:

(d) Someone other than the programmer should read the code, because the person who writes a particular subprogram may get into a mental rut and be unable to evaluate its correctness; this procedure is sometimes called a "structured walkthrough".

(e) Use a "trace", where the state of the simulation system, e.g. event list, state variables, certain statistical counters etc., is printed out after each event. This can also be used to check the model's reaction to extreme conditions.

(f) Run the model under simplifying assumptions if possible, for which the model's true characteristics are known or can be calculated.

In addition to the techniques listed above, the author adds the following two techniques from experience and exploitation of visual interactive system capabilities respectively:

(g) A few long run times are useful to check for unusual cases, rare bugs, instability, or an unusual or catastrophic combination of conditions.

(h) Careful observations of key events on the screen, "stepping" through each event in turn, and checking the attributes and movements of key entities (an animated equivalent of the trace, but not a complete replacement).

7.2 Verification of this model

Although the original model had been debugged, verified and validated for its original application, extensive verification was still required on the experimental model because of the major revisions made to most of the subroutines.

Functional testing was therefore required to ensure correct operation of the main capabilities described in Chapter 6, namely:

movement of jobs between machines

processing a job on a machine

handling breakdowns and repair times

choosing the next job for an idle machine

choosing the next operation for a job

Functional testing was carried out on a dummy data set, using FCFS despatching rules, and without alternative routes, and used most of the techniques listed in the previous section.

The third stage tested the action of the code handling despatching rules and alternative routes.

7.3 Components of the validation process

Validation is often poorly treated in the literature on simulation modelling, and where consideration has been given to validation it often relies extensively on historical data of existing systems or data collection on the real system (e.g. 82, 83). Much of the work carried out by the Simulation Group within the Manufacturing Systems Engineering Group at Warwick University has been involved with the design of production facilities not yet in existence and which are expected to be considerably different from their predecessor facilities. Modelling there proceeds using raw design data for input but output must be verified as "sensible" or "not sensible" by a team of experienced engineers and then "acceptable" or "of concern" or "unacceptable" subsequently.

Sensitivity analysis can be used to identify factors on which the results are very dependent and to check effects of some of the least reliable input data, such as expected breakdown levels or forecast product mix.

Since simulation depends on random number generation to activate events such as arrivals and breakdowns, any effects due to the random number streams being used must be found, before any effect due to a change in conditions can be quantified. The effects of random number streams will be considered in a later section (7.4) on "output analysis".

One advantage that a modeller has over other "experimental" conditions is much better "control" of the system, i.e. variables, factors and conditions can be changed easily by changing the data, changing the progam code or using other random number streams (84).

For the purposes of validation, the choice of probability functions to generate arrivals, time between breakdowns and repair times should be briefly justified.

Time between arrivals of each component type is sampled from a negative exponential distribution, which produces entirely "random" arrivals. It is questionable whether arrivals really are random in practice since they are likely to have departed from a preceding process or have arrived as a result of an order on a supplier. Both of these cases might indicate that a normal distribution would have been more appropriate.

The usual alternative is to assume that there is an unlimited supply of material. Parts are then drawn through the system as required, which creates another set of problems such as work in progress and inflated loading of a system which was not intended to run at full capacity.

It is therefore common to use the negative exponential distribution for arrivals and then prioritise or route the arrivals according to some schedule or final demand pattern. The pattern of time between arrivals in this model is shown by the graph in Figure 21 for components 1 to 4. The typical exponential curve is achieved with many short inter-arrival times but also a few longer times.

Similarly, the time between failures is sampled from a negative exponential distribution in order to generate "random" breakdown times.

Finally, it is common practice to generate repair times from an Erlang distribution which is a variant of the Gamma distribution. Of all the theoretical distributions commonly applied to simulation this application appears to be the most obvious. Depending on the type of equipment, it is likely that the repair time pattern would be made up of a lot of short breakdowns with a few long ones, or many long and complicated repairs with some short ones. These cases can be generated by specifying different "N" values. Graphs of the repair times produced using "shape factors" of N = 1, 2 and 3 are shown in Figure 22, generated from the data in table 2. This is important to scheduling rules, and particularly to alternative routes since different patterns could dictate different rule emphasis. With increasing data collection on machine history, it is not difficult in many companies to actually identify the repair pattern. In one company (79), analysis of breakdown data over the previous 5 years, showed similar patterns. However, an arbitrary decision was made to adopt Erlang with a value of N = 2.

7.4 Output analysis

The following sections will address important issues in dealing with the output from the model:

(i) Starting conditions and initial transient behaviour

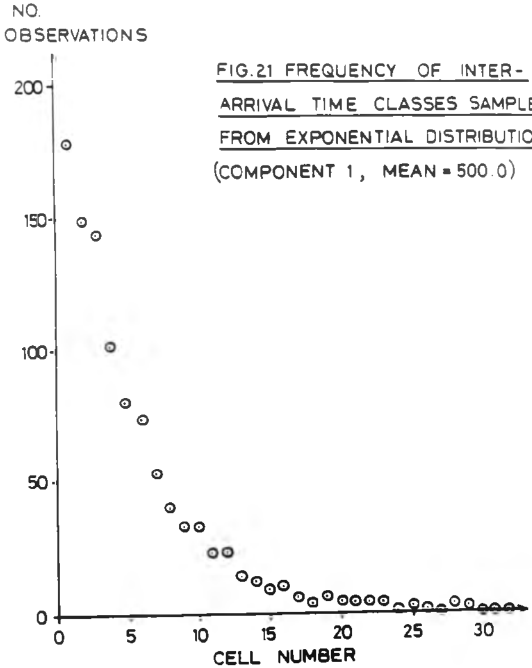


FIG.21 FREQUENCY OF INTER-ARRIVAL TIME CLASSES SAMPLED FROM EXPONENTIAL DISTRIBUTION (COMPONENT 1, MEAN = 500.0)

Table 2 Classification of repair times sampled from Erlang distribution with mean of 100 minutes

Cell No.	Range		No. observations		
			n=1	n=2	n=3
1	0 ≤ t <	20	65	21	8
2	20	40	58	43	38
3	40	60	41	56	58
4	60	80	45	52	56
5	80	100	28	50	66
6	100	120	42	36	44
7	120	140	18	31	32
8	140	160	15	20	27
9	160	180	9	21	16
10	180	200	7	10	9
11	200	220	7	6	8
12	220	240	9	9	6
13	240	260	6	6	2
14	260	280	1	7	1
15	280	300	6	1	2
16	300	320	2	1	1
17	320	340	5	1	0
18	340	360	2	1	0
19	360	380	5	1	0
20	380	400	8	0	1
21	400	420	0	0	1
22	420	440	0	1	0
23	440	460	0	0	0
24	460	480	0	0	0
25	480	500	0	0	0

FIG. 22a OBSERVED DISTRIBUTION OF REPAIR TIMES SAMPLED FROM ERLANG DISTRIBUTION – N = 1

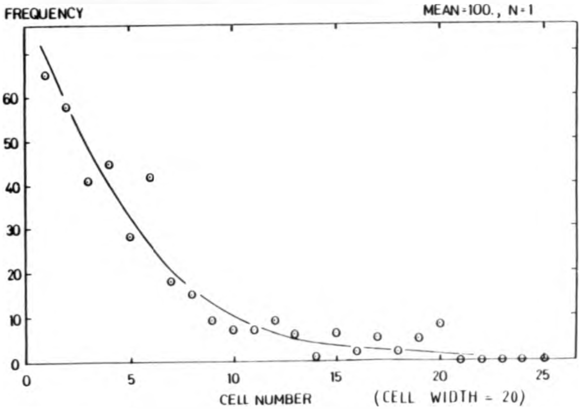


FIG. 22b OBSERVED DISTRIBUTION OF REPAIR TIMES SAMPLED FROM ERLANG DISTRIBUTION – N = 2

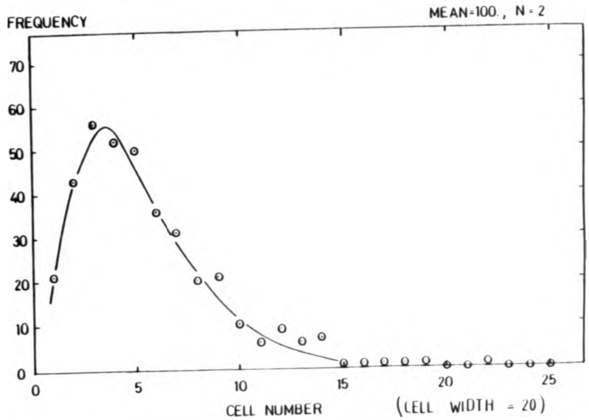
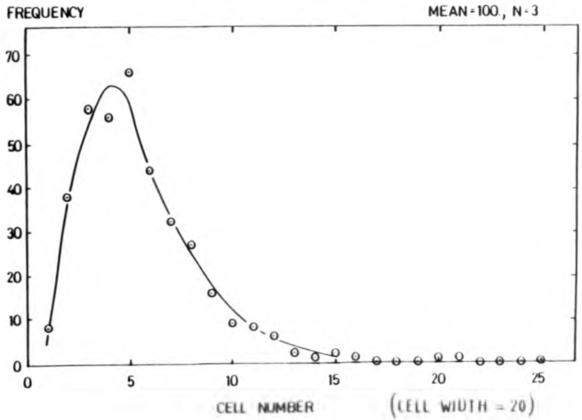


FIG. 22c OBSERVED DISTRIBUTION OF REPAIR TIMES SAMPLED FROM ERLANG DISTRIBUTION – N = 3



(ii) Choosing a representative run period

(iii) Variation in results due to random number streams

(iv) Assessing changes in results due to changed conditions

7.5 Starting conditions and transient behaviour

There are two common ways of starting simulations:

(i) From "cold", with no entities in the system and all facilities idle. Results from this "run in" period, or "warming up" period are ignored.

(ii) Using "typical" conditions as a starting point. In manufacturing simulation, this involves loading in a typical work-in-progress level, and a typical condition in which machines or operating facilities might be found, e.g. some part-finished jobs, some machines broken down, some idle or being set up.

Clearly, using method (i), there will be transient behaviour until the WIP level has built up in the system. Method (ii) is used to reduce the length of the transient period. Inevitably some bias will result from either method (82, 83, 84). In both methods, the same starting conditions should be used for all experiments, though this may not be appropriate in all cases under method (ii), e.g. where different rules being used could significantly affect the WIP levels .

Method (ii) could have been used here by running the model until steady state behaviour has been achieved and then transferring those conditions to the starting point of all successive experients. Method (i) was chosen in order to allow each set of conditions to settle into their own steady state and then take measurements of that steady state behaviour.

The end of transient behaviour must be determined. The system is considered to be in a steady state if its behaviour is independent of the starting conditions, and if the "probability of being in one of its states is governed by a fixed probability function" (84), e.g. machine working 40% of time available.

A number of techniques can be used to assess the length of the run-in period. Component flow time was chosen as the primary indicator or response variable and following Welch (85), moving averages of flow times were taken across varying numbers of periods. The choice of period length was arbitrary and results are discussed in the next section.

Since this method was inconclusive, a cumulative average flow time was calculated at the end of each period, for all parts finished since the start of the simulation, such that effects of early periods should become decreasingly important as steady state behaviour becomes more dominant. This technique has been used to good effect on other models by the author.

7.6 Representative run time period

Having established the duration of transient behaviour, it is necessary to determine the length of run required to obtain satisfactory results. There is a balance between time-wasting long runs and confidence in the results.

For this task, moving averages of results are the obvious analytical tool. The number of periods taken in the moving average is increased until the variation between moving averages is acceptable. The definition of "acceptable" is clearly variable, but another factor in simulation is required here. Different random number streams may generate different results even over quite long periods.

7.7 Effect of random number streams

Using any random number stream generates sampling errors. Pidd (84) quotes Saliby (86) who distinguishes two types of error:

(i) The "set" effect due to the set of values produced by the sampling process. There will never be perfect correspondence between the sample distribution generated and the theoretical distribution (e.g. figure 21).

(ii) The "sequence" effect due to the sequence in which the set of values is produced.

These errors are addressed in practice by using "variance reduction" techniques. In one method, the model is run several times under identical conditions, but using different random number streams, called replication. Thus a number of different runs should be carried out, and the greater the number of runs, the greater will be the confidence in the estimate of the true value of the response variable.

In another method, runs are made under new and old conditions, but using "common random numbers", i.e. the same random number streams are dedicated to the same activities under both sets of conditions.

A third alternative is to use antithetic streams, but it appears to be generally agreed (82, 83, 84) that analysis of these results is difficult and this method should be viewed with caution. In general, variance reduction techniques are used to increase the accuracy of the response variables, to reduce the run time necessary to achieve the same confidence in the results, and to assist in comparison between conditions. They do address the "set" effect, but cannot address the "sequence" effect.

7.8 Comparing results from runs under different conditions

In most manufacturing simulations, where results are collected for different entities, those results are "auto-correlated", that is, they are not independent. For example, waiting time depends on other members of the queue who are also waiting. Independent results are necessary in order to carry out statistical analysis on the results.

Auto-correlation can be addressed in three ways:

(i) Replication - repeating the run under the same conditions using a different random number stream achieves an independent result for each run.

(ii) Batching - Divide the results of a long single run into segments such that each segment contains the same number of observations. It is to be hoped that batches would be independent but in practice they are still likely to be auto-correlated, unless consecutive batches are separated by intervals whose results are discarded.

(iii) Regenerative methods - In some models, identical conditions can occur from time to time, e.g. in a queuing model where all servers are idle and all queues are empty. This is called a regeneration point and the time between regeneration points is known as an epoch. The results of activities from different "epochs" should be independent of one another. However, in this model it was considered unlikely that a regeneration point would occur, and the time between any regeneration points would be potentially too great for consideration except at very low loading levels.

For use on this model, replication appears to be the most suitable technique because it ensures independent results of similar sample sizes within a reasonable and reproducible period of time.

To prevent the need for very complicated statistical analysis, it is preferable that only one factor is different between any two sets of conditions being compared (84).

7.9 Simple data set used to test validation procedure

A simple data set was chosen to test the validation procedure in order to make any effects very visible, and to enable a high chance of correlation between static analysis and simulation results, thus testing the goodness of the model. However, the complexity of validation is also well demonstrated by even this simple data set.

There are four machines and four components. Each component follows the same production route comprising three operations. Each machine is capable of processing each component. Only one machine is available for the first operation, and one machine for the last operation, but two machines are capable of the second operation and they both draw work from the same buffer (i.e. pooled buffer).

Figure 23 shows the layout and machine capabilities. Table 3 shows the order of operations, operation codes and operation times. It can be seen that the single outer machines (in groups 1 and 4) are the rate limiters. The loading level is controlled by the mean inter-arrival time. Other conditions are:

No breakdowns

Set up times are all 30 minutes

No travel time between operations

First come, first served despatching rule

No alternative routes

An arbitrary period time of 2500 time units, which are nominal minutes and will henceforth be referred to as minutes, was chosen and runs of 45 periods carried out. Since 2500 minutes is 41.67 hours, it may be regarded as a one shift working week with a small amount of overtime, and 45 periods are therefore approximately 45 weeks, or perhaps a 48 week year of 39 hours per week.

Figure 23 Configuration of machines used for validation

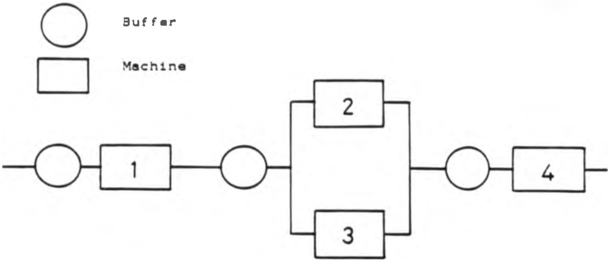


Table 3 Operation data used for validation

Component	Op. no.	Op. time and capable machines 1	2	3	4
1	1	1.00			
	2		1.25	1.25	
	3				1.00
2	1	1.50			
	2		1.88	1.88	
	3				1.50
3	1	1.80			
	2		2.25	2.25	
	3				1.80
4	1	2.50			
	2		3.12	3.12	
	3				2.50

7.10 Series A - Response at different loading levels

A loading level of 40% is taken to mean that a machine will be "busy" (i..e. working and not setting up) for an average of 40% of the time available. The calculation of inter-arrival times is shown in Table 4.

Other loading levels, using the same operation data can simply be generated by factoring the mean inter-arrival time used for 40% loading, e.g. 75% loading demands a mean inter-arrival time of 40/75 of the 40% mean inter-arrival time, and 20% loading demands 40/20 of the 40% inter-arrival time. Table 5 shows the mean inter-arrival times used for different loading levels (plotted in Figure 24).

An initial graph showing the average flowtime over 25 consecutive periods against the loading level yielded the familiar curve in Figure 25. It was concluded that build up of WIP and associated lengthening of queues and increase in flowtime were in accordance with expectations.

7.11 Series B - Achieving stable output

However, the average flowtime plotted on Figure 25 was an average from periods 4 - 28 (i.e. 25 periods, after an arbitrary allowance of three periods for warming up), and the results for different periods varied significantly. Figure 26 shows the variation in average flow time per period for component 1 by period for periods 1 - 45.

Table 4 Calculation of inter-arrival times

(a) 40% busy time in 2500 minute period is 1000 minutes

(b) Assume that components contribute equally to loading

(c) Batch size of 50 components

To achieve 40% loading on machine 1:

	a	b	c	d
Comp No.	Machine load (mins)	No. comps in period (a/op. time)	No. batches in period (b/50)	Inter-arrival time (mins) (2500/c)
1	250	250	5.0	500
2	250	167	3.3	750
3	250	139	2.78	900
4	250	100	2.0	1250
Total	1000			

Table 5 Inter-arrival time sets for different loading levels on machine 1

Code in Fig 24	Loading level (machine 1)	Inter-arrival time in minutes Comp 1	Comp 2	Comp 3	Comp 4
E	20%	1000	1500	1800	2500
D	40%	500	750	900	1250
C	50%	400	600	720	1000
B	60%	333	500	600	833
A	70%	267	400	480	667

FIG. 24 UTILISATION OF MACHINE 1 ARISING FROM DIFFERENT SETS OF INTER-ARRIVAL TIMES (SETS A TO E)

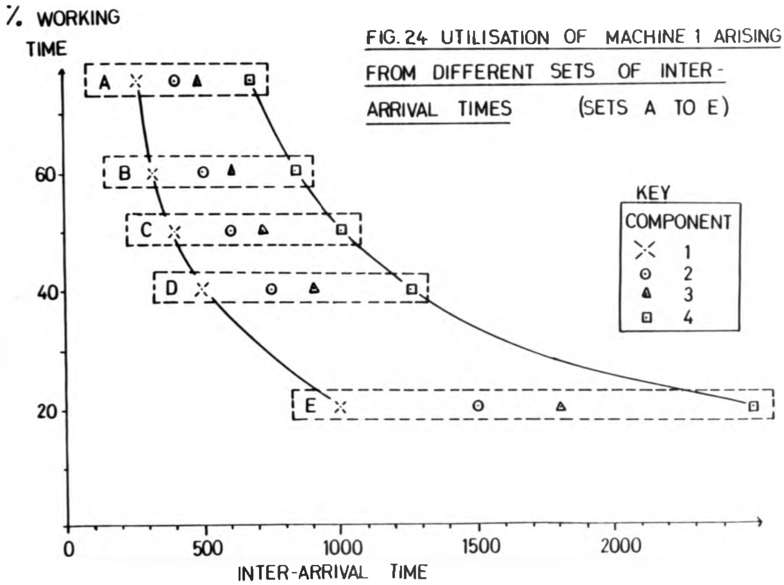


FIG. 25 OBSERVED FLOWTIME VS. MACHINE UTILISATION (MACHINE 1)

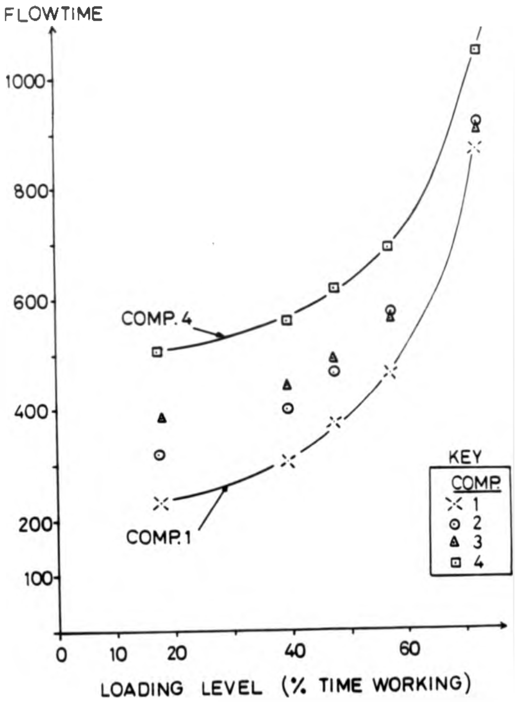
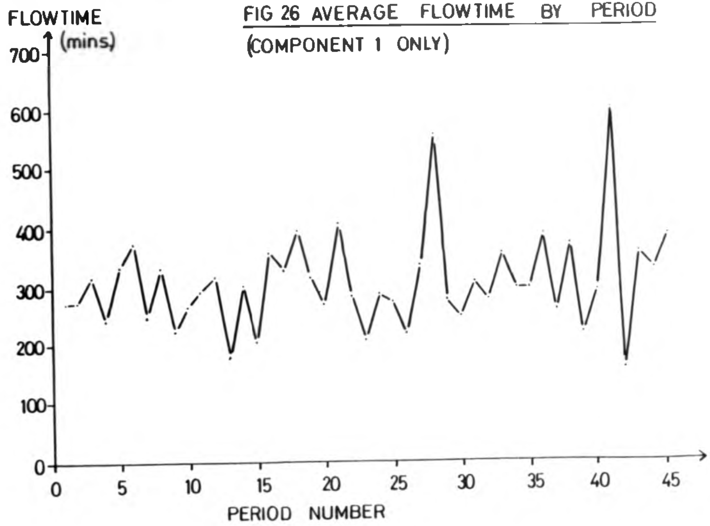


FIG 26 AVERAGE FLOWTIME BY PERIOD

(COMPONENT 1 ONLY)



As described in the previous section, moving averages can be used to gain a clearer picture of trends in results:

$$\text{Moving average} = \frac{1}{m} \quad \frac{\sum_{i}^{i+m-1} \bar{f}_i \cdot n_i}{\sum_{i}^{i+m-1} n_i}$$

where:

m = number of periods in moving average

i = period number

$\bar{f}_i$ = mean flowtime in period i

n_i =number of pallets completed in period i

Figure 27 shows moving averages for these results for component 1 over 15, 20, 25 and 30 periods. As the number of periods increases, it becomes clear that the flowtime is increasing. This could be due to either an effect of the random number stream being used, or an indication that the model is creating entities, and thus gradually filling the system itself, or that there is a very long and very slow warming up period.

Careful checks were made on the entities joining and leaving the system and it was concluded that further entities were not being created inside the system.

The next stage of testing examined the effect of random number streams and figure 28 shows the results for component 1 using 5 different streams for inter-arrival times, taking moving averages over 30 periods, and also a grand mean of the results for each group of periods.

FIG 27 MOVING AVERAGES OF FLOWTIME RESULTS
NP = NO. PERIODS IN MOVING AVERAGE COMPONENT 1

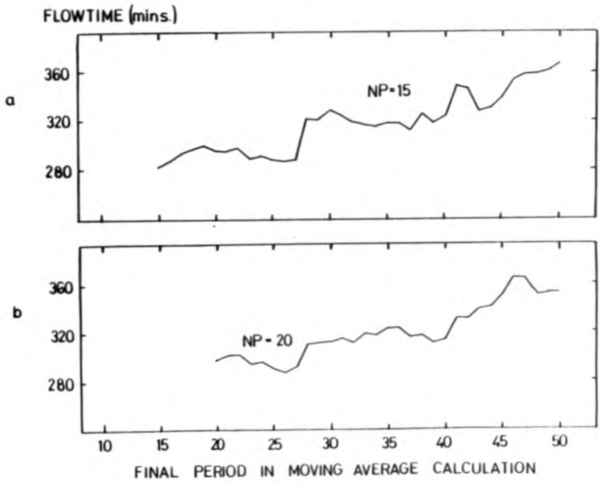


FIG 27 CONTINUED

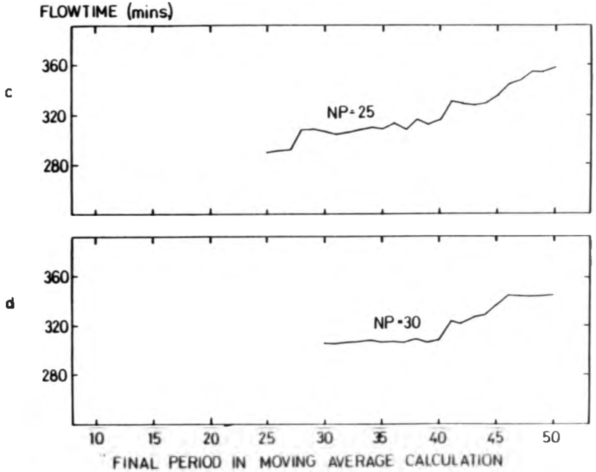


FIG.28 COMPARISON OF FLOWTIME MOVING AVERAGE RESULTS FROM DIFFERENT RANDOM NUMBER STREAM SETS FOR COMPONENT 1
(MOVING AVE OF 30 PERIODS, PERIOD = 2500 MINS)

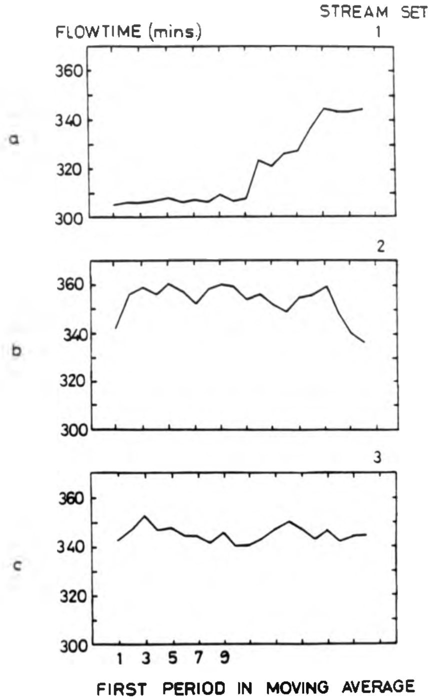
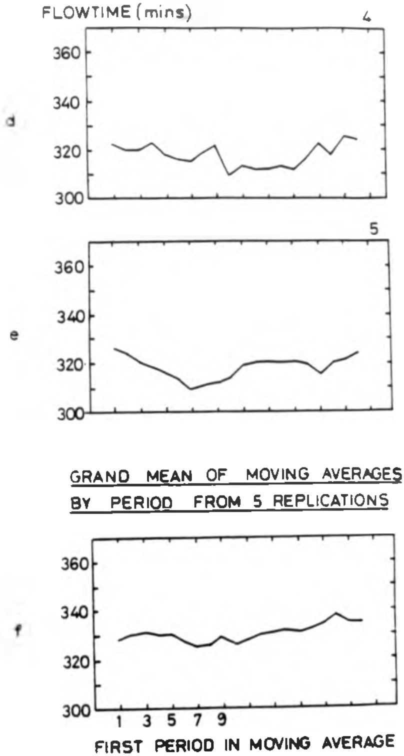


FIG. 28 CONTINUED



The grand mean of moving averages was calculated using:

$$\text{GMMA} = \frac{1}{r} \; \frac{1}{m} \; \frac{\sum_{1}^{r}\left[\sum_{i}^{i+m-1} \bar{f}_{ri} \, n_{ri}\right]}{\sum_{1}^{r} \sum_{i}^{i+m-1} n_{ri}}$$

where:

GMMA = Grand mean of moving averages

r = number of replications

i = period number

m = number of periods in moving average

$\bar{f}_{ri}$ = mean flowtime in period i of replication r

n_{ri} = number of pallets completed in period i of replication r

Comparing the results from stream set 1 and stream set 3 for example, it can be seen that there is a significant variation in the trend from different stream sets for the same component.

These averaged results, i.e. the grand mean of the five results for each moving average period group, were produced for each of the other 3 components, and are shown in Figure 29. From Figure 29, it appears that the results cannot be judged as "stable". Although the results for component 2 appear reasonably stable, the flowtime trend for components 1 and 3 and possibly 4 could be increasing. Cumulative average flowtime results were similarly inconclusive.

The experiments were repeated using a period length of 10,000 minutes and run again for 50 periods. Figure 30 shows averages of the moving average period

FIG. 29 GRAND MEANS OVER 5 REPLICATIONS OF 30 PERIOD MOVING AVERAGES (ALL COMPS, PERIOD = 2500 MINUTES)

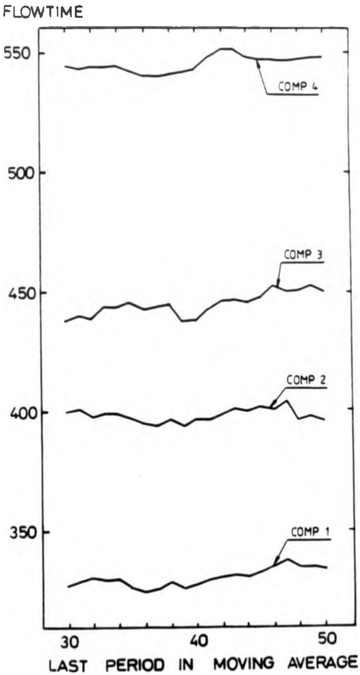
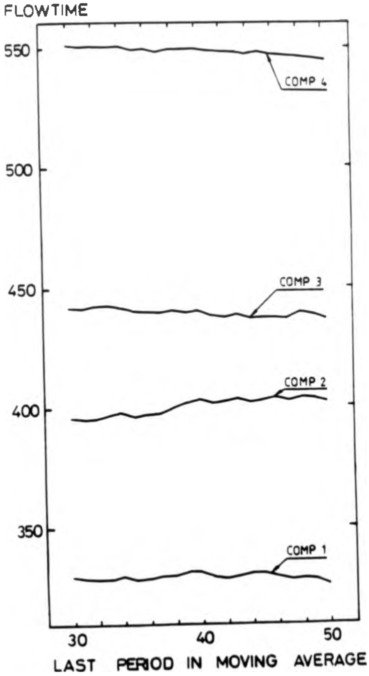


FIG. 30 GRAND MEANS OVER 5 REPLICATIONS OF 30 PERIOD MOVING AVERAGES (ALL COMPS, PERIOD = 10,000 MINUTES)



group constructed in a similar manner to those in Figure 29. It can be seen that the flow times for component 1 are steady, for component 2 are increasing and for components 3 and 4 are decreasing.

Instability in a model of a manufacturing application usually arises because the rate of arrival of parts exceeds the rate of processing and work in progress builds up without relief. Alternatively, programming errors can cause entities to be lost or invented falsely. The entity life cycles had already been checked so decreasing average flow times did not indicate an abnormal loss of work in progress.

Under stable running conditions, some periodic fluctuations in flow times are likely due to the effect of different random number streams. The model was therefore judged to be stable, and that a long run time, of the order of 50 periods each of 10,000 minutes is necessary to achieve this.

7.12 Series C - Effect of more disruptive data

The data set used so far is not very "disruptive". The results showed that component flow time comprised approximately 25% queuing time and 75% working time. In practice, the flowtime can be 20 or 30 times the sum of the batch processing times or hundreds of times the sum of the individual operation times.

The inter-arrival times of the simple data set were factored to achieve about 65% machine working time and breakdowns were introduced at a level of 10% of working time, i.e. about 6-7% of total time available. Using the previous results as a starting point, e.g. period length of 10,000 minutes, and following the same methodology, this data set was used to establish

(i) length of warm up period

(ii) length of run

(iii) number of replications

7.12.1 Length of warm up period

Table 6 shows moving averages of component flow time over 10 and 20 periods for one replication. Although the first few lines of each table appear to show increasing moving average results, it may be seen that lines 16-20 in each table contain even lower results as a consequence of the interaction of random events during those periods. It was concluded that the model warms up very quickly, probably within the first period, but that results from the first 5 periods would be discarded to be safe.

7.12.2 Length of run and number of replications

Twenty replications of 100 periods were made and were analysed in groups of five replications. Small Fortran programs were written to calculate moving averages and grand means of component flow time and machine working time over 5, 10, 15 and 20 replications.

Since the objective was to identify a combination of run length and number of replications where the variation in results was "acceptable", the output was analysed in two ways:

Table 6(a) Use of moving average results to estimate length of "warm up" period

Moving averages over [illegible] periods

Pd	Component 1		Component 2		Component 3		Component 4	
	MAVE	NOBS	MAVE	NOBS	MAVE	NOBS	MAVE	NOBS
1	982.0	[illegible]	1123	116	1311.3	164	1427	[illegible]
2	1072.0	346	1136.3	219	1421.7	170	1133.7	134
3	[illegible]	349	1230.0	218	1437.0	181	1335	140
4	1195.0	349	1004.6	222	1545.2	180	1327	145
5	1440.0	350	1023.6	[illegible]	1776.3	187	1730	140
6	1515.3	373	1642.3	219	1372.3	196	1334	138
7	1383	347	1667	223	1899.0	199	1392.2	140
8	1554.1	346	1616.7	233	1355.1	201	1366.1	133
9	1455.0	44	1542.2	235	1786.6	196	1700	129
10	1436.3	344	1526.5	238	1737.4	188	1775.1	126
11	1373.2	351	1506.3	226	1666.7	197	1781.4	124
12	1306.0	342	1449.6	227	1599.3	188	1621.	121
13	1335.4	335	1487.6	227	1628.4	184	1655.5	125
14	1269.6	339	1421.2	229	1544.1	179	1521.4	127
15	1020.6	328	1214.2	217	1325.1	173	1395.4	133
16	880.3	313	1069.8	218	1171.5	187	1211.0	133
17	847.5	311	1034.2	205	1093.0	174	1192.7	133
18	895.2	318	1069.7	197	1118.1	182	1201.0	131
19	959.9	311	1167.4	205	1181.2	189	1230.3	139
20	985.3	315	1163.1	205	1177.5	191	1265.9	145

where:

PD = period number

MAVE = moving average flowtime (in nominal minutes)

NOBS = number of observations in moving average calculation

Table 6(b) Moving averages over 20 periods

Format as in table 6(a)

Pd	Component 1		Component 2		Component 3		Component 4	
1	1182.8	684	1322.0	442	1505.5	361	1564.0	249
2	1188.3	688	1325.2	446	1515.2	358	1575.1	255
3	1226.4	684	1361.4	445	1533.5	365	1592.1	265
4	1232.0	688	1363.8	451	1544.7	359	1578.1	272
5	1240.0	687	1371.7	442	1559.4	360	1592.6	273
6	1225.7	686	1356.9	437	1550.2	363	1554.1	271
7	1235.8	658	1364.0	428	1522.5	373	1546.6	273
8	1238.5	664	1366.1	430	1504.9	383	1542.2	269
9	1220.4	655	1367.6	440	1489.4	385	1511.8	268
10	1221.0	659	1358.3	443	1480.0	379	1493.4	271
11	1206.3	658	1350.4	434	1456.4	380	1484.3	270
12	1165.8	655	1321.1	437	1406.4	374	1446.6	256
13	1149.4	652	1316.0	431	1392.2	366	1432.3	257
14	1114.2	649	1273.8	429	1341.8	358	1383.6	253
15	980.4	646	1149.5	421	1212.0	361	1303.2	255
16	929.8	637	1111.7	432	1165.0	360	1243.9	251
17	944.3	638	1117.3	421	1159.7	368	1251.5	250
18	971.5	643	1143.5	425	1190.6	370	1275.9	252
19	967.4	640	1142.7	421	1181.8	370	1269.4	253
20	946 0	650	1121.8	423	1175.1	363	1255.4	252

Table 6(c) Moving averages over 30 periods

Pd	Component 1		Component 2		Component 3		Component 4	
1	1131.4	991	1276.8	650	1412.8	544	1466.7	395
2	1133.4	1001	1279.4	656	1411.2	544	1476.5	390
3	1139.8	1001	1287.1	649	1407.0	547	1469.0	397
4	1142.6	998	1284.3	651	1409.9	538	1472.5	398
5	1144.8	1005	1279.3	646	1404.5	549	1472.2	395
6	1148.2	1010	1290.4	651	1414.5	556	1471.3	389
7	1169.6	985	1307.7	644	1418.8	567	1478.1	390
8	1175.3	989	1311.0	658	1424.5	571	1484.8	390
9	1138.2	984	1285.8	656	1391.2	566	1431.7	382
10	1115.9	994	1267.5	661	1384.0	551	1422.0	378
11	1107.5	1004	1264.2	653	1378.5	552	1425.2	372
12	1073.0	998	1235.6	653	1337.1	544	1387.6	363
13	1076.6	989	1249.5	659	1334.7	535	1393.3	368
14	1071.8	1004	1237.3	664	1312.2	533	1367.3	363
15	985.4	994	1163.3	663	1229.3	526	1327.4	368
16	929.1	988	1108.6	674	1170.3	519	1269.3	360
17	933.4	1004	1112.6	677	1173.3	529	1276.0	356
18	933.1	1005	1116.2	669	1170.3	535	1269.5	364
19	958.1	1012	1134.9	671	1184.8	542	1289.1	367
20	960.8	1007	1136.3	673	1193.3	543	1290.7	373

(a) A ratio was constructed for the highest grand mean moving average flowtime result divided by the lowest, in that output file, for each component. An example of one result is shown in Table 7 for five replications and a moving average of 30 periods. The highest values observed are roughly 20% higher than the lowest value observed. An example of a similar process on machine working time is shown in Table 8, where the differences are about 2-3%. The ratio results for different moving average groups and numbers of replications are shown in Tables 9 and 10.

(b) 95% confidence intervals were calculated for each combination of moving average groups and numbers of replications, assuming that flowtimes conformed to a Normal distribution. One example of this output is shown in Table 11 and Table 12 contains an example for machine utilisation. Results for component flow time and machine working time were collated to Tables 13 and 14 respectively.

Examination of Tables 9, 10, 13 and 14 indicates that flow time is much more variable than machine working time. From the table of flow time ratios, there is little benefit from using 50 or 60 periods to construct moving averages rather than 40 periods. Similarly, 15 or 20 replications do not improve the results from 10 replications (in fact, the results from 20 replications were worse in all cases). Values shown in table 14 of machine utilisation ratios do not conflict with these conclusions.

From the table of 95% confidence limits for flowtimes, it is observed that there is some improvement in taking 50 rather than 40 periods in the moving average, but little benefit in taking 60 periods. Similarly, it appears that there is little benefit

Table 7 Grand means of moving averages of flowtimes over 5 replications - origin of ratio to be used in stability check

Pd	Component 1		Component 2		Component 3		Component 4	
1	1175.8	5004.	1343.4	3279.	1510.1	2745	1574.4	1398.
2	1190.6	4970	1357.1	3270.	1520.9	2758.	1584.9	2012.
3	1199.5	4987.	1365.4	3280.	1529.5	2764.	1593.4	2024
4	1167.1	4986	1334.8	3275.	1494.0	2742.	1552.7	2010
5	1146.3	4975.	1314.7	3270.	1477.6	2719.	1535.6	2009
6	1140.6	4997.	1309.5	3271.	1471.1	2711.	1533.5	1992
7	1126.4	4971.	1290.9	3262.	1455.3	2711.	1526.3	1978
8	1112.0	4943.	1275.0	3243.	1434.4	2720.	1511.8	1969.
9	1100.3	4988	1259.9	3241.	1416.8	2713.	1494.0	1969
10	1064.5	5000.	1224.6	3226.	1376.3	2704.	1466.6	1983.
11	1043.3	4969.	1201.9	3227.	1357.0	2690.	1450.5	1981
12	1035.7	5001.	1187.1	3265.	1344.2	2664.	1434.0	1976.
13	1017.1	4998.	1168.1	3261.	1323.1	2666.	1401.4	1975.
14	1022.8	5033.	1163.3	3265.	1301.6	2663.	1403.3	1979
15	1020.7	5010.	1163.8	3286.	1297.1	2664	1398.4	1981
16	1023.0	5033	1171.3	3302.	1301.4	2635.	1408.8	1981
17	1015.6	5036.	1162.0	3325.	1297.4	2635.	1393.6	1967
18	1012.9	5058.	1151.7	3301.	1288.3	2631.	1384.9	1961
19	1010.7	5040.	1147.3	3287.	1286.1	2638.	1387.3	1937
20	1007.7	5012.	1145.4	3314.	1283.5	2644.	1382.3	1914.
21	1008.7	5016.	1142.6	3320.	1283.6	2668.	1380.0	1395
22	1011.0	5040.	1144.0	3330.	1288.8	2662.	1385.9	1892.
23	1008.1	5026.	1145.2	3331.	1289.6	2664	1383.2	1990
24	1010.0	5014.	1141.6	3320.	1289.5	2677.	1388.6	1888
25	1023.4	4975.	1158.4	3328.	1311.8	2703.	1404.5	1391
26	1038.7	4994.	1178.3	3333.	1325.3	2721.	1417.7	1900
27	1031.6	4957.	1175.5	3349.	1320.9	2742.	1402.0	1911
28	1024.8	4940.	1167.8	3358.	1315.4	2719.	1391.2	1920
29	1022.1	4913.	1166.1	3364.	1312.8	2727.	1388.3	1912
30	1009.3	4942.	1151.4	3383.	1297.4	2713.	1379.1	1890
31	1009.0	4959.	1148.5	3359.	1292.8	2711.	1376.3	1904
32	1005.2	4963.	1149.2	3372.	1294.4	2702.	1377.4	1908
33	994.2	4956.	1136.8	3383.	1281.0	2708.	1369.3	1904
34	996.6	4927.	1136.7	3365.	1274.5	2724.	1368.2	1914
35	1001.9	4926.	1140.7	3367.	1277.6	2728.	1374.1	1922
36	997.1	4928.	1136.4	3331.	1271.9	2741.	1370.8	1921
37	1011.1	4939.	1149.3	3348.	1286.9	2743.	1375.7	1911
38	1037.4	4955.	1171.4	3373.	1317.8	2739.	1396.9	1919
39	1042.1	4925.	1181.6	3358.	1325.6	2761.	1407.0	1923
40	1044.9	4907.	1180.7	3358.	1325.6	2763.	1403.2	1906
41	1047.0	4931	1183.2	3371.	1325.5	2753.	1408.7	1917
42	1056.8	4922.	1193.9	3345.	1328.0	2769.	1417.7	1933
43	1067.9	4913.	1202.4	3329.	1344.9	2775.	1434.2	1938
44	1052.3	4888.	1189.8	3326.	1335.4	2778.	1420.8	1939
45	1042.1	4396.	1174.6	3282.	1317.3	2774.	1405.2	1926

Table 7 continued

Pd	Component 1		Component 2		Component 3		Component 4	
46	1059.3	4863.	1179.6	3267.	1335.4	2787.	1419.1	1940.
47	1067.3	4885.	1186.1	3260.	1339.5	2794.	1430.3	1957.
48	1064.8	4889.	1195.6	3265.	1340.2	2804	1427 5	1962.
49	1072.9	4882.	1192.2	3275.	1356.1	2307	1439.2	1980.
50	1081.2	4909.	1197.0	3274.	1365.8	2789.	1450.2	1989.
51	1087.3	4903.	1204.5	3275.	1371.3	2791.	1454.0	2011.
52	1095.6	4966.	1215.6	3257.	1387.5	2799.	1458.6	2020.
53	1093.3	4976.	1211.3	3276.	1387.3	2799.	1456.9	2006.
54	1078.0	4893.	1198.4	3273.	1370.7	2788.	1435.9	1996
55	1066.1	4917.	1179.4	3243.	1348.0	2770.	1416.5	1988.
56	1074.6	4911.	1184.2	3237.	1361.5	2775.	1431.3	1991.
57	1088.9	4913.	1201.1	3237.	1381.5	2796.	1444.7	1983.
58	1088.7	4922.	1199.4	3240.	1378.9	2794.	1444.2	1974.
59	1098.0	4902.	1208.5	3256.	1383.5	2311	1449 6	1974.
60	1108.1	4867.	1218.2	3234	1394.2	2810.	1452.7	1989.
61	1098.7	4952.	1207 1	3250.	1394.1	2307	1445.6	1974
62	1087.4	4853.	1194.3	3246.	1375.7	2793.	1429.9	1966.
63	1092.0	4834.	1202.1	3262.	1383.2	2786.	1434.9	1969.
64	1114.8	4851.	1219.2	3279.	1406.9	2774.	1455.0	1977
65	1122.4	4831.	1226.8	3278.	1411.8	2795.	1464 5	1979.
66	1126.8	4822.	1230.2	3286.	1413.9	2783.	1466.6	1997
67	1114.9	4823.	1221.5	3272.	1402.1	2771.	1453.7	2007.
68	1100.4	4807.	1211.1	3252.	1385.9	2757.	1451.7	2018.
69	1103.5	4831.	1213.0	3261.	1385.5	2740.	1448.2	2011.
70	1105.7	4801.	1220.1	3269.	1393.9	2741.	1448.1	2018.
71	1102.0	4802.	1211.6	3282.	1388.4	2751.	1443.2	2006.
72	0.0	0.	0.0	0.	0.0	0.	0.0	0.
73	0.0	0.	0.0	0.	0.0	0.	0.0	0.
74	0.0	0.	0.0	0.	0.0	0.	0.0	0.
75	0.0	0.	0.0	0.	0.0	0.	0.0	0.

	Component 1	Component 2	Component 3	Component 4
Max	1199.5	1365.4	1529.5	1593.4
Min	994.2	1136.4	1271.9	1368.2
Div	1.207	1.202	1.203	1.165

Max = maximum flowtime value observed in the column

Min = minimum flowtime value observed in the column

Div = maximum value divided by minimum value ("ratio")

Table 8 Grand means of 30 period moving averages of machine 1 working time over 5 replications - origin of ratio to be used in stability check

Period	% Working time of machine no: 1	2	3	4
1	64.4	47.3	35.0	64.2
2	64.5	47.3	35.1	64.3
3	64.6	47.3	35.2	64.6
4	64.3	47.2	35.0	64.2
5	64.2	47.1	34.8	64.0
6	64.0	47.0	34.8	63.9
7	63.8	46.9	34.7	63.7
8	63.5	46.7	34.6	63.4
9	63.6	46.8	34.6	63.5
10	63.6	46.8	34.6	63.5
11	63.4	46.7	34.5	63.3
12	63.5	46.7	34.5	63.4
13	63.5	46.7	34.5	63.3
14	63.7	46.9	34.7	63.4
15	63.7	46.8	34.6	63.5
16	63.8	46.9	34.6	63.5
17	63.8	46.9	34.6	63.5
18	63.6	46.8	34.5	63.4
19	63.3	46.7	34.3	63.2
20	63.1	46.5	34.2	63.0
21	63.3	46.6	34.3	63.1
22	63.3	46.6	34.3	63.1
23	63.2	46.5	34.4	63.0
24	63.2	46.5	34.5	62.9
25	63.3	46.6	34.4	63.0
26	63.6	46.9	34.5	63.4
27	63.7	46.8	34.6	63.5
28	63.6	46.8	34.5	63.4
29	63.5	46.8	34.5	63.3
30	63.4	46.8	34.5	63.2
31	63.5	46.7	34.6	63.2
32	63.5	46.8	34.5	63.3
33	63.5	46.8	34.5	63.2
34	63.5	46.8	34.7	63.3
35	63.6	46.9	34.7	63.3
36	63.6	46.9	34.8	63.2
37	63.7	46.8	34.7	63.3
38	63.9	47.0	34.8	63.5
39	63.9	46.9	34.8	63.5
40	63.7	46.8	34.7	63.3
41	63.8	46.9	34.7	63.5
42	64.0	47.0	34.9	63.6
43	63.9	47.0	34.8	63.6
44	63.7	46.9	34.7	63.5
45	63.5	46.7	34.5	63.2

Table 8 continued

Period	% Working time of machine no: 1	2	3	4
46	63.6	46.8	34.6	63.1
47	63.8	47.0	34.6	63.4
48	63.9	47.1	34.7	63.5
49	64.2	47.2	34.9	63.7
50	64.2	47.3	34.9	63.8
51	64.3	47.3	35.0	63.9
52	64.3	47.3	35.0	63.9
53	64.2	47.3	34.9	63.9
54	64.0	47.1	34.8	63.8
55	63.9	47.1	34.7	63.6
56	63.8	46.9	34.8	63.6
57	63.9	47.0	34.9	63.7
58	64.0	46.9	34.9	63.7
59	64.1	47.0	35.0	63.8
60	64.0	47.0	34.3	63.7
61	63.3	46.9	34.7	63.5
62	63.6	46.7	34.7	63.3
63	63.6	46.8	34.7	63.4
64	63.3	46.9	34.8	63.5
65	63.9	46.8	34.9	63.6
66	63.9	46.9	34.8	63.7
67	63.8	46.9	34.8	63.6
68	63.6	46.8	34.7	63.5
69	63.6	46.7	34.8	63.4
70	63.6	46.6	34.8	63.4
71	63.5	46.6	34.9	63.3
72	0.0	0.0	0.0	0.0
73	0.0	0.0	0.0	0.0
74	0.0	0.0	0.0	0.0
75	0.0	0.0	0.0	0.0
Max	64.6	47.3	35.2	64.6
Min	63.1	46.5	34.2	62.9
Div	1.023	1.018	1.029	1.026

Max = maximum value observed in the column

Min = minimum value observed in the column

Div = maximum value divided by minimum value ("ratio")

Table 9 Summary of ratios observed from 5, 10, 15 and 20 replications over moving average lengths of 1, 20, 30, 40, 50 and 60 periods (period = 10,000 minutes) for component flowtime

Comp	NREPS	No. periods in moving average					
		1	20	30	40	50	60
1	5	2.691	1.304	1.207	1.115	1.085	1.065
	10	2.135	1.129	1.118	1.048	1.052	1.034
	15	2.109	1.085	1.108	1.044	1.045	1.024
	20	1.987	1.133	1.153	1.070	1 056	1.037
2	5	2.365	1.301	1.202	1.114	1.080	1.071
	10	1.964	1.127	1.113	1.045	1.048	1.037
	15	1.905	1.077	1.098	1.038	1.039	1.027
	20	1.891	1.118	1.135	1.061	1.047	1.035
3	5	2.620	1.299	1.203	1.124	1.094	1.078
	10	1.952	1.132	1.113	1.055	1.053	1.038
	15	1.869	1.076	1.091	1.041	1.042	1.023
	20	1.832	1.106	1.121	1.062	1.047	1.030
4	5	2.352	1.242	1.165	1.103	1.070	1.059
	10	2.060	1.099	1.095	1.040	1.041	1.029
	15	1.941	1.061	1.084	1.038	1.034	1.025
	20	1.865	1.096	1.115	1.056	1.043	1.028

where NREPS = number of replications

Table 10. Summary of ratios observed from 5, 10, 15 and 20 replications over moving average lengths of 1, 20, 30, 40, 50 and 60 periods (period = 10,000 minutes) for machine working time

Machine	NREPS	No. periods in moving average					
		1	20	30	40	50	60
1	5	1.225	1.035	1.023	1.016	1.010	1.008
	10	1.201	1.019	1.015	1.009	1.009	1.006
	15	1.146	1.021	1.009	1.008	1.007	1.008
	20	1.134	1.026	1.015	1.010	1.009	1.007
2	5	1.238	1.041	1.018	1.016	1.011	1.007
	10	1.190	1.028	1.013	1.009	1.008	1.006
	15	1.165	1.027	1.009	1.007	1.006	1.005
	20	1.143	1.027	1.013	1.012	1.011	1.008
3	5	1.321	1.038	1.029	1.017	1.014	1.010
	10	1.282	1.023	1.016	1.013	1.012	1.008
	15	1.233	1.018	1.010	1.010	1.007	1.007
	20	1.171	1.028	1.018	1.013	1.009	1.009
4	5	1.241	1.042	1.028	1.018	1.012	1.009
	10	1.238	1.024	1.014	1.009	1.008	1.007
	15	1.161	1.017	1.012	1.008	1.004	1.005
	20	1.149	1.025	1.018	1.011	1.010	1.007

where NREPS = number of replications

Table 11 Example of output from program which calculates mean, standard deviation and 95% confidence limits for component flowtimes - Results for component 1

	No. periods in moving average:					
	1	20	30	40	50	60
	5 replications					
Mean	1329.6	1183.1	1140.4	1101.1	1095.5	1069.1
S.D.	342.9	38.2	32.3	28.0	17.0	23.1
C.L.	951.8	105.9	89.7	77.8	47.3	64.2
	10 replications					
	1139.7	1037.7	1046.0	1018.4	1029.4	1013.3
	250.3	55.7	40.4	35.3	27.1	26.7
	566.2	125.9	91.3	79.7	61.2	60.4
	15 replications					
	1013.3	1021.7	1045.6	1016.2	1027.6	1014.2
	214.2	55.2	38.4	33.8	25.6	24.0
	459.5	118.4	82.4	72.5	55.0	51.4
	20 replications					
	976.0	1070.6	1090.5	1048.2	1048.0	1038.9
	194.9	56.8	40.9	35.4	26.7	25.1
	407.9	118.8	85.6	74.1	55.3	52.5

where:
S.D. = standard deviation
C.L. = value of 95 % confidence statistic

Table 12 Example of output from program which calculates mean, standard deviation and 95% confidence limits for machine utilisation (working time only) - Machine 1

No. periods in moving average:

	1	20	30	40	50	60
5 replications						
Mean	68.6	64.6	64.0	64.3	64.0	63.8
S.D.	2.8	0.1	0.3	0.2	0.3	0.3
C.L.	7.3	0.3	0.8	0.7	0.9	0.7
10 replications						
	66.4	63.0	63.1	63.3	63.2	63.1
	2.4	0.6	0.4	0.4	0.4	0.3
	5.5	1.3	0.9	0.9	0.9	0.7
15 replications						
	63.5	63.0	63.2	63.2	63.1	63.1
	2.5	0.6	0.4	0.4	0.4	0.3
	5.4	1.2	0.8	0.8	0.8	0.6
20 replications						
	62.9	63.7	63.8	63.7	63.6	63.5
	2.6	0.6	0.4	0.4	0.4	0.3
	5.5	1.3	0.9	0.9	0.8	0.7

where:
S.D. = standard deviation
C.L. = value of 95 % confidence statistic

Table 13 Summary of 95% confidence test statistics from 5, 10, 15 and 20 replications over moving average lengths of 1, 20, 30, 40, 50 and 60 periods (period = 10,000 minutes) for component flowtime

Comp	NREPS	No. periods in moving average					
		1	20	30	40	50	60
1	5	952	108	90	78	47	64
	10	566	126	91	80	61	60
	15	460	118	82	73	55	52
	20	408	119	86	74	56	53
2	5	2575	695	506	439	326	309
	10	1090	301	218	189	141	134
	15	747	213	151	132	99	92
	20	597	178	125	110	82	76
3	5	3610	1036	742	649	482	449
	10	1468	427	306	267	200	185
	15	970	287	203	178	133	123
	20	750	226	161	141	105	97
4	5	4446	1319	942	828	613	567
	10	1778	525	375	330	243	226
	15	1157	343	242	214	157	145
	20	884	264	185	164	120	112

where NREPS = number of replications

Table 14 Summary of 95% confidence test statistics from 5, 10, 15 and 20 replications over moving average lengths of 1, 20, 30, 40, 50 and 60 periods (period = 10,000 minutes) for machine working time

Machine	NREPS	No. periods in moving average					
		1	20	30	40	50	60
1	5	7.8	0.3	0.8	0.7	0.9	0.7
	10	5.5	1.3	0.9	0.9	0.9	0.7
	15	5.4	1.2	0.8	0.8	0.8	0.6
	20	5.5	1.3	0.9	0.9	0.8	0.7
2	5	32.3	7.3	5.3	5.3	4.8	3.9
	10	12.9	2.9	2.1	2.1	1.9	1.6
	15	8.5	1.9	1.3	1.3	1.2	1.0
	20	6.6	1.5	1.1	1.0	1.0	0.8
3	5	38.5	8.5	6.2	6.1	5.5	4.6
	10	15.0	3.4	2.4	2.4	2.2	1.8
	15	9.5	2.1	1.5	1.5	1.4	1.1
	20	7.2	1.7	1.2	1.2	1.1	0.9
4	5	42.5	9.6	6.9	6.8	6.2	5.0
	10	17.1	3.9	2.8	2.8	2.5	2.1
	15	11.4	2.6	1.8	1.8	1.7	1.4
	20	9.0	2.1	1.5	1.5	1.3	1.1

where NREPS = number of replications

from making 20 replications rather than 15, although 15 replications yielded tighter results than 10 replications.

At this point, some arbitrary judgement of acceptability is required. There are two practical considerations:

(a) It is quicker and easier to increase the length of the simulation run than it is to increase the number of replications.

(b) Results from any set of conditions will be compared to a set of base conditions, using common random number streams, and analysed by paired comparison. The use of common random number streams will further reduce the variance.

It is therefore concluded that 10 replications of 50 periods should be made, using a period length of 10,000 minutes.

Using the inter-arrival times for the "disruptive" data set which creates about 65% working time on machine 1, this run time yields the following effective sample sizes:

Component 1 Sample size $= \dfrac{50 \times 10{,}000}{307} = 1628$

Component 2 Sample size $= \dfrac{50 \times 10{,}000}{462} = 1082$

Component 3 Sample size $= \dfrac{50 \times 10{,}000}{554} = 902$

Component 4 Sample size $= \dfrac{50 \times 10{,}000}{769} = 650$

These sample sizes are considered to be very adequate to yield average flow times for each replication.

7.13 Random Number Stream Management

Only random number streams used for arrival time generation have been varied so far. Random number streams are also used:

(i) to generate breakdown times

(ii) to generate repair times

(iii) to make routing decisions (random rule only)

The purpose of using random number streams is to generate interactions between activities that are likely to interact in
to next breakdown for a machine is linked to time worked, and time worked depends on the sequence of arrivals which is varying with different random number stream sets, it will not be necessary to vary breakdown time generation for each replication. A similar argument for repair times depends on different arrival sequences causing breakdowns at different times and hence the repair times on different machines will not always interact in the same way.

This argument could be extended to the random number stream used for making routing decisions, but it was decided to vary this on each run to ensure variations.

7.14 Output Distribution

The output distributions of inter-arrival times and repair times have been recorded, and the output distribution of flow times was similarly recorded (see fig.31). It is observed to be right hand skewed, with the right side of the maximum appearing to be similar to a Normal distribution shape.

In fact the output distribution is less important when replications are made. If mean flowtime results are recorded from each replication, test statistics for confidence intervals can be easily derived. The Central Limit Theorem states that the means of samples from a population will be Normally distributed. Hence, the test statistics for results in Section 7.12 were derived using Student's t distribution:

$$\bar{x}(n) \pm t_{n-1,\,0.975}\, s(n)$$

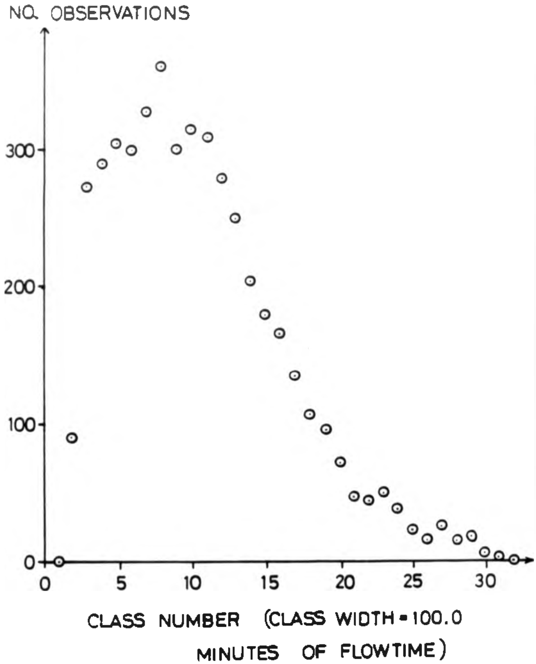
where:

$\bar{x}(n)$ = mean of results from n replications

$s(n)$ = standard deviation of results from n replications

$t_{n-1,.975}$ = value from t-distribution table for n-1 degrees of freedom, where upper critical point = 0.975

FIG. 31 DISTRIBUTION OF OBSERVED FLOWTIMES (ALL COMPONENTS)



7.15 Effect of Despatching Rules

It had been intended to test the effect of different despatching rules on the performance of the alternate route decision rules. Because the model has been set up to achieve a certain utilisation level rather than draw in new jobs as required (Section 6.9), the overall level of work in progress is low, and different despatching rules were found to make negligible difference to performance. It is the case that despatching rules are used for queue management and only queues of minimal average size were generated. Without queues of waiting work from which jobs must be chosen, the effect on job progress caused by using despatching rules is negligible. In practice, queues may arise for many reasons but it may be argued that work in progress is often unnecessarily high, and causes undue focus on despatching rules. This will be discussed further in Chapter 10.

It is anticipated that despatching rules would have some impact on the performance of alternate routing rules at higher levels of machine utilisation and work in progress, and indeed could alter the apparent relative performance of different rules. It is recognised that flexible manufacturing systems are characterised by high utilisation from economic necessity and that a high work in progress level is frequently a feature of a batch production machine shop. It may be argued that this low level of work in progress is unrealistic.

It is also expected that there would be interactions between some despatching rules and some routing rules. Before interactions are examined, it is wise to understand the underlying effects of different rules.

Because despatching rules did not appear to generate significantly different results at this level of machine utilisation, and because these conditions allow the underlying effects of the routing rules to be examined as a first stage of investigating alternate routing, it was decided to eliminate despatching rules altogether and maintain this level of utilisation. Any queues which do form are managed by the first come, first served rule.

8. Single operation alternatives

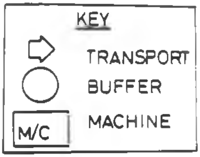
Single operation alternatives will be examined first in order to learn about the performance of the decision rules before applying them to partial routes.

8.1 Machine and buffer configurations

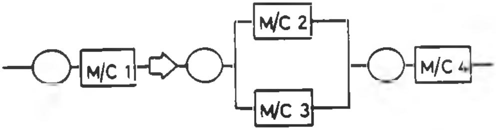
Different configurations for single operation alternatives are shown in Figure 32(a), (b) and (c). Dispersed buffers represent the case where the alternative machines are located sufficiently far apart to need separately designated areas for work awaiting processing, perhaps in different bays or different shops. The material mover would need to know the destined machine before transportation starts. Inter-operational transportation will also occur after operations at machines 2 or 3, to reach machine 4, but this movement is not shown in the diagram in order to highlight the different transport, buffer and decision sequences after machine 1.

Figure 32(b) shows schematically a "shared" buffer, where two machines are located sufficiently close to one another to draw material from the same floor stock. In the diagram, each job in the shared buffer has been nominally allocated to one machine or the other. To distinguish this case from the "first free machine" decision sequence, consider for example the ratio rule described in Section 5.3, where jobs are allocated to the routes and hence to the alternative machines rather than being drawn by the machines. Therefore, jobs are allocated to one of the two machines on arrival at the floor stock or input buffer, and the buffer is depicted in Figure 32(b) as two adjacent queues.

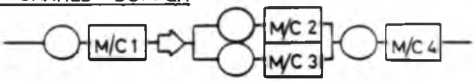
FIG 32 SCHEMATIC REPRESENTATION OF CONFIGURATIONS WITH SINGLE OPERATION ALTERNATIVES



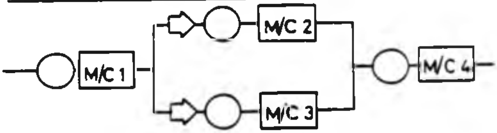
A FIRST FREE MACHINE



B. SHARED BUFFER



C. DISPERSED BUFFERS



Where machines draw work from a shared buffer, i.e. the first free machine will draw work, the allocation decision is made as the job is being removed from the buffer, shown in Figure 32(a).

One further arrangement was tested. In this case, alternate routes were not available and components 1 and 3 were always routed to machine 2, and components 2 and 4 were always routed to machine 3. The purpose of this arrangement was to act as a comparator. While still requiring set up changes on each machine, the number of set up changes may be expected to be less than the more flexible arrangements and should therefore allow shorter flowtimes.

8.2 Description of rules

The following rules were tested on the shared buffer and dispersed buffer configurations.

8.2.1 "Random" - A random number is selected from a uniform distribution between 0 and 1. For a decision between two machines or routes, a random number of 0.5 or less results in a choice of route 1, and greater than 0.5 results in a choice of route 2.

8.2.2 "Ratio" - Each route maintains a counter of jobs already allocated. The counters are compared to the ratio of jobs expected for each route, and the job is directed to the route with the least achieved ratio. In the event of a tie, the first-found least achieved ratio is accepted, which would be route 1 if there are are only two routes.

8.2.3 "Number of jobs" - The job is directed to the machine or route which has the least number of jobs waiting at the time of the decision. No account is taken of jobs which may be in transit to join the queue. In the event of a tie, the job is directed to the first examined queue, i.e. route 1.

8.2.4 "Least work in next queue (LWINQ)" - The job is directed to the machine or route where the queue of waiting jobs represents the least work load for the machine. In the event of a tie, the job is directed to the first examined queue, i.e. route 1.

8.2.5 "Earliest expected start (EES)" - In addition to the workload waiting in the queue, the completion time of the current job is added, plus the duration of any breakdown observed to be occurring at the time of the decision. The job is routed to the machine where it may be expected to be started first. In the event of a tie, the job is directed to route 1.

Rules 1 and 2 are non-adaptive. Rules 3, 4 and 5 are adaptive. Taking workload into account in manufacturing networks usually involves provision for effects of the despatching rule in operation. The rules as described here assume first come first served despatching and are tested in this way to obtain data on their broad effects on material flow.

8.3 Series 1 - Dispersed buffers

Ten replications of each of the 5 decision rule conditions were made, using run times of 55 periods at 10,000 minutes. Results from the first 5 periods were discarded according to the conditions established during validation. A typical summary report from one replication is shown in Appendix 5. The data and conditions used are shown in Table 15. Machines 2 and 3 are identical.

A small Fortran program was written to read the summary reports from each set of replications to calculate the mean, standard deviation and 95% confidence limits of the mean of the replication means. A sample of output from this program is shown in Appendix 7.

From the results observed during validation and from the summary of scheduling practice in chapter 4, it was recognised that the most important measure of success of the alternate routing rules is component flow time. Further insight can be gained into how the rule is operating by observing the relative working and setting times on the alternate machines 2 and 3. Reduced setting time should be reflected in reduced average flow times. Work in progress levels will be lower with lower flow times. The summary report (Appendix 5) also shows a breakdown of flowtime into proportions spent queuing, being processed, waiting during a breakdown and time travelling. Since processing time and travelling time are constant for any job, and time waiting during breakdowns is a function of the working time and is relatively reproducible in the long term, flowtime breakdown results mainly indicate the changing proportion of time spent queuing and therefore add no more information to the flowtime results themselves.

Table 15 Operation data used for series 1 and 2

Component	Op. no.	Op. time and capable machines 1	2	3	4
1	1	1.00			
	2		1.25	1.25	
	3				1.00
2	1	1.50			
	2		1.88	1.88	
	3				1.50
3	1	1.80			
	2		2.25	2.25	
	3				1.80
4	1	2.50			
	2		3.12	3.12	
	3				2.50

Other conditions:

Set-up times:	30 minutes
Inter-operational transport time:	30 minutes
Batch size:	50
Mean time to repair:	100 minutes
Mean time between failures:	13.2 hours

Notes:

Outer machines are bottlenecks

Expected bottleneck machine utilisation is 65%

Results for these performance measures for each of the five rules are presented in Tables 16 and 17.

8.3.1 Machine use

The two non-adaptive rules balance the workload across the two identical machines, 2 and 3, and setting time is also balanced. However, there is a significant reduction in flow time by adopting the ratio rule (see Table 16) and the ratio rule also increases the setting time on both machines. At this point, an item of data must be clarified. In order to maintain a feasible machine utilisation level and feasible setting up time proportion, minimum batch quantities have been specified for components 1 and 2 when being processed by machines 1 and 4. It should also be noted that machines 1 and 4 are more heavily loaded than machines 2 and 3 and hence machines 2 and 3 are not the bottleneck. Machines 1 and 4 are forced to process three consecutive batches of component 1 before being allowed to change a set up and 2 batches of component 2 before a set up change.

Considering the rule description in 8.2.2, the ratio rule operates in a strict "round robin" manner in its route allocations. The random rule allows periodic consecutive allocations to take the same route which may have coincided with consecutive batches of similar component type, removing the need for a set up change on the alternative route. The forced minimum batch quantities may have enhanced the probability that consecutive batches of a similar component could be routed to the same machine under the random rule.

It may be seen from Table 16 that the adaptive rule 3, examining the number of jobs in the queue, has no effect on the flow times. The interesting result is the

Table 16 Series 1: Dispersed buffers

Flowtime results

Rule	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	1074	1213	1386	1444
	St. Dev.	27	31	36	30
	U. C. L.	1135	1283	1467	1512
	L. C. L.	1013	1144	1306	1376
2. Ratio	Mean	1060	1195	1359	1420
	St. Dev.	24	27	31	27
	U. C. L.	1114	1255	1429	1481
	L. C. L.	1006	1135	1289	1358
3. No. jobs	Mean	1067	1206	1388	1443
	St. Dev.	25	28	32	30
	U. C. L.	1125	1270	1461	1511
	L. C. L.	1010	1142	1315	1374
4. LWINQ	Mean	1063	1200	1382	1432
	St. Dev.	26	27	33	31
	U. C. L.	1121	1260	1458	1502
	L. C. L.	1004	1139	1307	1362
5. EES	Mean	1045	1187	1347	1437
	St. Dev.	26	29	33	28
	U. C. L.	1104	1252	1421	1499
	L. C. L.	985	1123	1274	1375

Table 17 Series 1: Dispersed buffers

Machine use results

Rule	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
1. Random	Mean	40.3	8.6	41.0	8.7
	St.Dev.	0.3	0.1	0.3	0.1
	U.C.L.	40.9	8.8	41.7	8.8
	L.C.L.	39.7	8.4	40.3	8.5
2. Ratio	Mean	40.7	10.2	40.7	10.3
	St.Dev.	0.2	0.0	0.2	0.0
	U.C.L.	41.0	10.3	41.0	10.4
	L.C.L.	40.3	10.1	40.3	10.2
3. No.jobs	Mean	61.2	13.0	20.1	4.4
	St.Dev.	0.1	0.0	0.2	0.1
	U.C.L.	61.5	13.1	20.6	4.6
	L.C.L.	60.8	12.8	19.6	4.3
4. LWINQ	Mean	61.2	12.9	20.1	4.4
	St.Dev.	0.2	0.0	0.2	0.1
	U.C.L.	61.6	13.0	20.6	4.6
	L.C.L.	60.9	12.8	19.6	4.2
5. EES	Mean	46.3	10.7	35.0	8.2
	St.Dev.	0.1	0.0	0.2	0.0
	U.C.L.	46.6	10.8	35.5	8.3
	L.C.L.	46.0	10.7	34.5	8.1

imbalance of workload between the two alternate machines 2 and 3 in table 17, and associated difference in setting up time, also seen in the results for rule 4. It is considered that this imbalance is due to the number of ties which are broken by taking route 1. A "tie" is used to denote the case where each alternate appears equally good according to the decision criteria e.g. both alternate machines have the same number of jobs waiting. In the event of a tie, the first-found machine which meets the "least" condition will be the default route. For only two routes this will always be the machine on route 1. If there were more routes, and the routes were always examined in the same order, the routes examined earlier would always tend to attract more jobs if ties are broken in this way. Hence, to achieve load balancing if it were required, these rules should operate with a second rule used to break ties, such as the round robin rule to make a decision based on the way that previous ties have been broken.

8.3.2 Flowtime

The "least work in next queue" rule did not improve the flow time results from the random rule. The only adaptive rule to achieve improvement was the "earliest expected start" rule. It is to be expected that rules which take account of more local network information would perform better. This rule has two pieces of information more than rule 4, namely state of the job currently being processed (if any) and remaining time to the end of the current breakdown (if any). The remaining processing time for a job which has just started could vary from 62 minutes for a pallet of component 1 to 156 minutes for a pallet of component 2. Reference to Figure 22(b) of the Erlang repair times indicates that breakdowns could possibly be up to 400 minutes, although the mean is 100 minutes, and all processing is halted during the repair. The frequency of breakdowns should be considerably less than the frequency of batches being processed. Although further work to examine the effect of each factor and its relative importance is required to be certain, it is proposed that the knowledge of current processing times is more critical.

8.3.3 Rule comparisons

Tables 18 and 19 show the 95% confidence limits for some selected comparisons of flow time and machine results from which some of the previous statements regarding significant reductions in flowtimes were made. The tables have been constructed by matching replications from different rules but which used the same random number stream set, i.e. common random numbers, in order to reduce the variance due to the random number streams still further. For example, "2 - 1" indicates that results for rule 1 were subtracted from each corresponding replication of results for rule 2. The mean, standard deviation and 95% confidence limits for the mean, were calculated on these differences using the method stated previously. Since the confidence intervals for flowtimes of all components under rules 3 and 4 contain zero, it was considered that there was no improvement by using these rules instead of the random allocation. Negative results at both limits for rule 5 compared to the random allocation indicate that the flowtime results were significantly lower when job processing time and repair times were taken into account.

8.3.4 Flowtime spread

Another important performance measure is the spread of flow times achieved by each rule. The maximum and minimum flow times observed for each component during the 50 period run time were recorded for each replication. The mean, standard deviation and 95% confidence limits for these maximum and minimum

Table 18 Series 1: Dispersed buffers

Comparisons of selected flowtime results

Rules	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
2 - 1	Mean	-13.7	-18.5	-27.4	-24.7
	St. Dev.	5.6	8.1	9.2	6.4
	U.C.L.	-1.1	-0.1	-6.5	-10.3
	L.C.L.	-26.3	-36.9	-48.3	-39.1
3 - 1	Mean	-6.7	-7.5	2.0	-1.7
	St. Dev.	7.5	6.6	8.0	6.1
	U.C.L.	10.3	7.4	20.0	12.1
	L.C.L.	-23.7	-22.4	-16.0	-15.5
4 - 1	Mean	-11.3	-13.9	-3.9	-12.2
	St. Dev.	6.6	7.9	8.4	6.1
	U.C.L	3.6	3.9	15.1	1.6
	L.C.L.	-26.2	-31.7	-22.9	-26.0
5 - 1	Mean	-29.2	-26.1	-38.9	-7.3
	St. Dev.	7.1	7.6	9.9	10.2
	U.C.L.	-13.1	-8.9	-16.5	15.9
	L.C.L	-45.3	-43.3	-61.3	-30.5
5 - 2	Mean	-15.5	-7.8	-11.5	17.4
	St. Dev.	6.6	7.3	6.2	9.1
	U.C.L.	-0.6	8.9	2.5	38.1
	L.C.L.	-30.4	-24.1	-25.5	-3.3

Table 19 Series 1: Dispersed buffers

Comparisons of selected machine results

Rules	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
2 - 1	Mean	0.3	1.7	-0.3	1.6
	St. Dev.	0.2	0.1	0.2	0.1
	U.C.L.	0.9	1.8	0.2	1.7
	L.C.L.	-0.2	1.5	-0.9	1.4
3 - 1	Mean	20.9	4.4	-20.9	-4.2
	St. Dev.	0.2	0.1	0.2	0.1
	U.C.L.	21.4	4.5	-20.3	-4.1
	L.C.L.	20.3	4.2	-21.4	-4.4
4 - 1	Mean	20.9	4.3	-20.9	-4.3
	St. Dev.	0.2	0.1	0.2	0.1
	U.C.L.	21.4	4.5	-20.4	-4.2
	L.C.L.	20.4	4.2	-21.4	-4.5
5 - 1	Mean	6.0	2.2	-6.0	-0.4
	St. Dev.	0.3	0.1	0.3	0.1
	U.C.L.	6.6	2.3	-5.3	-0.3
	L.C.L.	5.4	2.0	-6.6	-0.6
5 - 2	Mean	5.6	0.5	-5.6	-2.0
	St. Dev.	0.1	0.0	0.1	0.0
	U.C.L.	5.8	0.6	-5.4	-1.9
	L.C.L.	5.5	0.4	-5.8	-2.1

flowtime results are shown in Tables 20 and 21. The minimum flowtime results do not vary much between rules and were also found to vary minimally in all other series. They are therefore not presented again in other sections.

The maximum flowtimes for this series do not appear to change either and indeed a comparison of selected maximum flowtimes in Table 22 confirms this. It is also observed that the standard deviation of these maximum flowtimes remains similar for each rule.

Table 20 Series 1: Dispersed buffers

Maximum flowtimes recorded over 50,000 minutes for each rule:

Rule	Measure	Maximum flowtime			
		Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	3463	3577	3693	3709
	St. Dev.	172	179	152	155
	U. C. L.	3853	3982	4037	4060
	L. C. L.	3073	3172	3348	3357
2. Ratio	Mean	3434	3457	3631	3676
	St. Dev.	185	189	167	150
	U. C. L.	3852	3885	4008	4015
	L. C. L.	3015	3029	3254	3337
3. No. jobs	Mean	3461	3550	3689	3707
	St. Dev.	173	167	166	169
	U. C. L.	3853	3927	4064	4089
	L. C. L.	3068	3173	3315	3326
4. LWINQ	Mean	3474	3517	3677	3732
	St. Dev.	176	155	165	165
	U. C. L.	3871	3868	4049	4106
	L. C. L.	3077	3166	3304	3358
5. EES	Mean	3480	3529	3667	3727
	St. Dev.	174	173	167	161
	U. C. L.	3873	3919	4046	4093
	L. C. L.	3087	3138	3289	3362

Table 21 Series 1: Dispersed buffers

Minimum flowtimes recorded over 50,000 minutes for each rule:

Rule	Measure	Minimum flowtime			
		Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	223	254	277	325
	St. Dev.	0	0	3	4
	U.C.L.	223	254	283	334
	L.C.L.	223	254	271	315
2. Ratio	Mean	223	259	279	337
	St. Dev.	0	3	3	5
	U.C.L.	223	267	287	347
	L.C.L.	223	252	272	327
3. No. jobs	Mean	223	257	283	327
	St. Dev.	0	2	4	5
	U.C.L.	223	262	293	338
	L.C.L.	223	252	274	317
4. LWINQ	Mean	223	260	279	326
	St. Dev.	0	3	4	5
	U.C.L.	223	267	288	336
	L.C.L.	223	252	270	316
5. EES	Mean	223	262	278	321
	St. Dev.	0	4	3	3
	U.C.L.	223	270	289	329
	L.C.L.	223	253	271	314

Table 22 Series 1: Dispersed buffers

Comparison of selected maximum flowtimes

Rules	Measure	Maximum flowtimes Comp 1	Comp 2	Comp 3	Comp 4
2 - 1	Mean	-29	-120	-62	-33
	St. Dev.	39	47	54	45
	U. C. L.	59	-14	60	69
	L. C. L.	-117	-226	-183	-136
3 - 1	Mean	-2	-27	-3	-1
	St. Dev.	72	46	38	48
	U. C. L.	160	77	84	106
	L. C. L.	-164	-131	-90	-109
4 - 1	Mean	11	-60	-16	23
	St. Dev.	54	62	43	52
	U. C. L.	133	79	81	142
	L. C. L.	-110	-199	-113	-95
5 - 1	Mean	17	-48	-25	19
	St. Dev.	66	42	57	38
	U. C. L.	166	46	103	103
	L. C. L.	-131	-143	-154	-66
5 - 2	Mean	47	72	36	52
	St. Dev.	66	42	61	54
	U. C. L.	-197	167	173	174
	L. C. L.	-104	-22	-101	-71

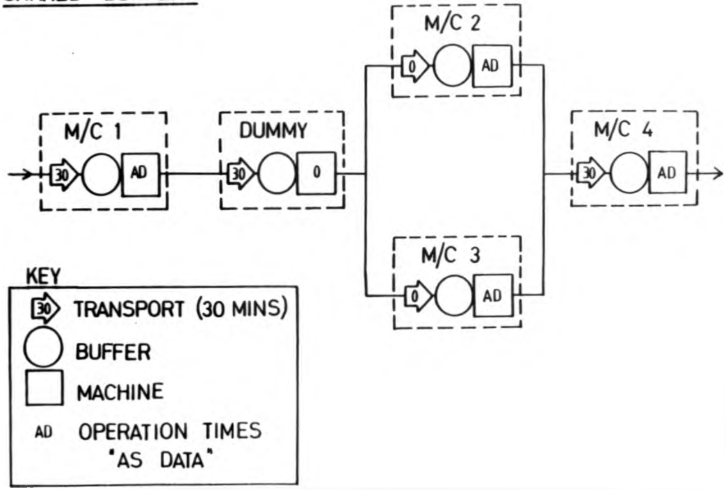
8.4 Series 2 - Allocated queues in shared buffer

The 5 decision rules were tested using allocated queues in a shared buffer. This configuration is not directly achievable in the model and a dummy machine is required.

The simulation model was intended to be a "generic" model, able to accept data from many different machine configurations. Each processing unit comprises a set of jobs in transit to the unit, a floor set of jobs which have arrived and are awaiting processing and a processor which may comprise one or more machines of identical or different capabilities. In preference to rewriting fundamental logic of the program and maintaining different versions of the program, and in order to allow a routing decision within a processing unit rather than outside or between processing units, a dummy processing unit was introduced between machine 1 and machines 2 and 3 as shown in Figure 33. The dummy unit has no processing time and no set up change time. After processing on the zero time operation, a routing decision is made between machines 2 and 3, and then a zero inter-operational transport time delivers the component to the floor stock of machine 2 or machine 3. This mechanism was verified by using the same data sets and random number stream sets, but zero inter-operational transport times throughout, for dispersed buffers and shared buffers. Identical results were not achieved immediately because of the way in which random number streams are allocated to machines to generate repair times. After amending this part of the program, identical results were achieved.

The 5 decision rules were then tested on the allocated buffers using the data and conditions shown in Table 15. Ten replications were made under each rule using

FIG 33 LOCATION OF DUMMY MACHINE REQUIRED TO ACHIEVE SHARED BUFFERS



the conditions described for series 1. The results were analysed in a similar manner. Tables 23 and 24 contain the mean, standard deviation and 95% confidence limits of the working and setting times of machines 2 and 3 and the flow times of all components. Tables 25 and 26 contain the mean, standard deviations and 95% confidence limits of the mean flow times and machine results for selected pair comparisons. Tables 27 and 28 compare the results for dispersed and shared buffers rule by rule.

All results for the random and ratio rules are identical. The random and ratio rules take no account of the state of the network. The same random number stream was used for routing decisions in each corresponding pair of replications producing an identical sequence of decisions. Similarly, the "round robin" ratio rule produced an identical sequence of decisions since the inter-operational transport time is the same for all jobs and at all stages. Since all other conditions, e.g. breakdown and repair time random number streams were the same, there were no other causes of variation in the flow time.

The results for the other rules show a small and insignificant variation in flow times. However, there is a difference between dispersed and shared buffers concerning the way in which the adaptive rules 3,4 and 5 make use of the alternative machines 2 and 3. In all 3 cases, less working and setting time is observed on machine 2 for dispersed buffers, and is compensated by increased use of machine 3.

The principal difference between the two configurations is the timing of the decision. It is suggested here that the average queue length at the time of the later decision on entry to the shared buffer will be less than the average queue length at

Table 23 Series 2: Allocated shared buffer

Flowtime results

Rule	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	1074	1213	1386	1444
	St. Dev.	27	31	36	30
	U. C. L.	1135	1283	1467	1512
	L. C. L.	1013	1144	1306	1376
2. Ratio	Mean	1060	1195	1359	1420
	St. Dev.	24	27	31	27
	U. C. L.	1114	1255	1429	1481
	L. C. L.	1006	1135	1289	1358
3. No. jobs	Mean	1068	1207	1383	1435
	St. Dev.	26	29	35	33
	U. C. L.	1127	1272	1463	1510
	L. C. L.	1009	1141	1304	1360
4. LWINQ	Mean	1067	1206	1383	1437
	St. Dev.	27	29	34	32
	U. C. L.	1127	1270	1460	1510
	L. C. L	1004	1141	1306	1364
5. EES	Mean	1042	1180	1345	1431
	St. Dev.	25	29	33	28
	U. C. L.	1097	1245	1418	1494
	L. C. L.	986	1115	1271	1367

Table 24 Series 2: Allocated shared buffer

Machine use results

Rule	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
1. Random	Mean	40.3	8.6	41.0	8.7
	St. Dev.	0.3	0.1	0.3	0.1
	U. C. L.	40.9	8.8	41.7	8.8
	L. C. L.	39.7	8.4	40.3	8.5
2. Ratio	Mean	40.7	10.2	40.7	10.3
	St. Dev.	0.2	0.0	0.2	0.0
	U. C. L.	41.0	10.3	41.0	10.4
	L. C. L.	40.3	10.1	40.3	10.2
3. No. jobs	Mean	63.8	13.3	17.5	3.6
	St. Dev.	0.2	0.1	0.2	0.0
	U. C. L.	64.2	13.5	18.0	3.7
	L. C. L.	63.4	13.2	17.1	3.5
4. LWINQ	Mean	63.8	13.3	17.5	3.5
	St. Dev.	0.2	0.1	0.2	0.0
	U. C. L.	64.2	13.5	17.9	3.6
	L. C. L.	63.4	13.2	17.1	3.5
5. EES	Mean	49.1	11.4	32.2	7.5
	St. Dev.	0.1	0.0	0.2	0.0
	U. C. L.	49.4	11.5	32.6	7.6
	L. C. L.	48.8	11.3	31.8	7.4

where: LWINQ = Least work in next queue
EES = Earliest expected start
U. C. L. = Upper limit of 95% confidence interval
L. C. L. = Lower limit of 95% confidence interval

Table 25 Series 2: Allocated shared buffer

Comparisons of selected flowtime results

Mean, standard deviation and 95% confidence limits were found for the differences between corresponding replication results. Refer to tables 23 and 24 for rule numbers.

Rules	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
2 - 1	Mean	-13.7	-18.5	-27.4	-24.7
	St. Dev.	5.6	8.1	9.2	6.4
	U.C.L.	-1.1	-0.1	-6.5	-10.3
	L.C.L.	-26.3	-36.9	-48.3	-39.1
3 - 1	Mean	-5.9	-6.7	-3.1	-9.4
	St. Dev.	7.5	7.6	8.8	8.5
	U.C.L.	11.0	10.5	16.7	9.8
	L.C.L.	-22.8	-23.9	-22.9	-28.6
4 - 1	Mean	-8.2	-7.7	-3.2	-7.2
	St. Dev.	5.5	6.9	8.0	6.9
	U.C.L.	4.2	8.0	14.8	8.5
	L.C.L.	-20.6	-23.4	-21.2	-22.9
5 - 1	Mean	-32.1	-33.7	-41.7	-13.6
	St. Dev.	5.4	5.9	6.7	5.5
	U.C.L.	-19.8	-20.3	-26.6	-1.1
	L.C.L.	-44.4	-47.1	-56.8	-26.1
5 - 2	Mean	-18.4	-15.2	-14.3	11.1
	St. Dev.	4.1	6.1	6.5	5.9
	U.C.L.	-9.2	-1.3	0.3	24.5
	L.C.L.	-27.6	-29.1	-28.9	-2.3

where, for example, "2 - 1" represents ratio results minus random results for each corresponding replication.

Table 26 Series 2: Allocated shared buffer

Comparisons of selected machine use results

Rules	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
2 - 1	Mean	0.3	1.7	-0.3	1.6
	St. Dev.	0.2	0.1	0.2	0.1
	U. C. L.	0.9	1.8	0.2	1.7
	L. C. L.	-0.2	1.5	-0.9	1.4
3 - 1	Mean	23.5	4.8	-23.5	-5.1
	St. Dev.	0.3	0.1	0.3	0.0
	U. C. L.	24.1	4.9	-22.8	-5.0
	L. C. L.	22.9	4.6	-24.1	-5.2
4 - 1	Mean	23.5	4.8	-23.5	-5.1
	St. Dev.	0.3	0.1	0.3	0.1
	U. C. L.	24.2	4.9	-22.9	-5.0
	L. C. L.	22.9	4.6	-24.1	-5.3
5 - 1	Mean	8.8	2.9	-8.8	-1.2
	St. Dev.	0.2	0.1	0.3	0.1
	U. C. L.	9.4	3.0	-8.2	-1.0
	L. C. L.	8.3	2.7	-9.4	-1.3
5 - 2	Mean	8.5	1.2	-8.5	-2.8
	St. Dev.	0.0	0.0	0.0	0.0
	U. C. L.	8.6	1.3	-8.4	-2.7
	L. C. L.	8.4	1.1	-8.6	-2.9

Table 27 Series 2: Comparison between flowtime results for shared and dispersed buffers, rule by rule

Rule	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	0.0	0.0	0.0	0.0
	St. Dev.	0.0	0.0	0.0	0.0
	U.C.L.				
	L.C.L.				
2. Ratio	Mean	0.0	0.0	0.0	0.0
	St. Dev.	0.0	0.0	0.0	0.0
	U.C.L.				
	L.C.L.				
3. No. jobs	Mean	-0.8	-0.8	5.1	7.7
	St. Dev.	7.1	7.2	7.5	8.0
	U.C.L.	15.2	15.5	22.1	25.8
	L.C.L.	-16.8	-17.1	-11.9	-10.4
4. LWINQ	Mean	-3.1	-6.2	-0.7	-5.0
	St. Dev.	5.4	6.1	5.6	5.9
	U.C.L.	9.2	7.6	12.0	8.3
	L.C.L.	-15.4	-20.0	-13.4	-18.3
5. EES	Mean	2.9	7.8	2.8	6.3
	St. Dev.	5.4	4.2	6.5	5.7
	U.C.L.	15.1	17.2	17.6	19.1
	L.C.L.	-9.3	-2.0	-12.0	-6.5

Table 28 Series 2: Comparison between machine use results for shared and dispersed buffers, rule by rule

Rule	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
1. Random	Mean	0.0	0.0	0.0	0.0
	St. Dev.	0.0	0.0	0.0	0.0
	U.C.L.				
	L.C.L.				
2. Ratio	Mean	0.0	0.0	0.0	0.0
	St. Dev.	0.0	0.0	0.0	0.0
	U.C.L.				
	L.C.L.				
3. No. jobs	Mean	-2.6	-0.4	2.6	0.9
	St. Dev.	0.1	0.0	0.1	0.1
	U.C.L.	-2.3	-0.3	2.9	1.0
	L.C.L.	-2.9	-0.5	2.3	0.7
4. LWINQ	Mean	-2.6	-0.4	2.6	0.8
	St. Dev.	0.1	0.0	0.1	0.1
	U.C.L.	-2.3	-0.3	2.9	1.0
	L.C.L.	-2.9	-0.5	2.3	0.7
5. EES	Mean	-2.8	-0.7	2.9	0.7
	St. Dev.	0.1	0.0	0.1	0.0
	U.C.L.	-2.7	-0.6	3.0	0.8
	L.C.L.	-3.0	-0.8	2.7	0.6

(Note: "dispersed" results minus "shared" results) the time of making a decision for dispersed buffers. The probability of "catching up" with the job in front while it is waiting in the next queue must decrease with time. Hence a job which is about to enter an allocated queue in the shared buffer is less likely to have to take other waiting jobs into account when making a decision.

A small amendment to the program was made to record the size of the queue for each alternate machine, at the time the routing decision is made, for both shared buffer and dispersed buffer configurations.

With dispersed buffers, 4745 decisions were made in the test period. The following shows the frequency of occurrence (i.e. number of observations) of different queue lengths at the time of the routing decision:

Queue length	Machine 2	Machine 3
0	3225	4431
1	1473	311
2	47	3

(There were no observations of queue length of 3 or more)

In the same period 4744 decisions were made with shared allocated buffers. The number of observations of different queue lengths was as follows:

Queue length	Machine 2	Machine 3
0	3486	4656
1	1245	88
2	13	0

(There were no observations of queue length of 3 or more)

These results show that queue lengths in this part of the system are small and confirm the inappropriateness of using despatching rules.

Jobs waiting for machine 2 are seen more frequently using dispersed buffers (1473 + 47), i.e. an earlier decision, than when using shared/allocated buffers (1245 + 13). Hence more jobs are likely to be sent to machine 3 using dispersed buffers. Therefore, inter-operational transport time affects the routing decision under certain buffer configurations.

Tables 29 and 30 show that the mean and standard deviation of maximum flowtimes change insignificantly with the change in decision rules.

Table 29 Series 2: Allocated shared buffer

Maximum flowtimes

Rule	Measure	Maximum flowtime			
		Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	3463	3577	3693	3709
	St. Dev.	172	179	152	155
	U. C. L.	3853	3982	4037	4060
	L. C. L.	3073	3172	3348	3357
2. Ratio	Mean	3434	3457	3631	3676
	St. Dev.	185	189	167	150
	U. C. L.	3852	3885	4008	4015
	L. C. L.	3015	3029	3254	3337
3. No. jobs	Mean	3484	3565	3726	3730
	St. Dev.	190	194	184	191
	U. C. L.	3913	4004	4143	4161
	L. C. L.	3056	3125	3309	3299
4. LWINQ	Mean	3470	3564	3726	3703
	St. Dev.	191	190	179	178
	U. C. L.	3903	3993	4131	4105
	L. C. L.	3037	3137	3320	3301
5. EES	Mean	3498	3520	3703	3715
	St. Dev.	163	184	163	155
	U. C. L.	3867	3936	4071	4065
	L. C. L.	3129	3103	3336	3365

Table 30 Series 2: Allocated shared buffer

Comparison of selected maximum flowtimes

Rules	Measure	Maximum flowtime			
		Comp 1	Comp 2	Comp 3	Comp 4
2 - 1	Mean	-29	-120	-62	-33
	St. Dev.	39	47	54	45
	U. C. L.	59	-14	60	69
	L. C. L.	-117	-226	-183	-134
3 - 1	Mean	22	-12	34	21
	St. Dev.	52	57	63	70
	U. C. L.	140	117	175	179
	L. C. L.	-97	-142	-108	-137
4 - 1	Mean	7	-14	33	-6
	St. Dev.	50	45	57	61
	U. C. L.	120	89	162	133
	L. C. L.	-105	-116	-95	-145
5 - 1	Mean	35	-58	11	6
	St. Dev.	81	43	66	49
	U. C. L.	218	41	161	117
	L. C. L.	-147	-156	-139	-105
5 - 2	Mean	64	63	72	39
	St. Dev.	82	38	52	62
	U. C. L.	249	148	190	180
	L. C. L.	-120	-23	-45	-101

8.5. Series 3 - Comparison with "first free" machine and fixed routing

Runs using the "first free" machine rule used the same data as series 1 and 2, shown in Table 15. The operational data used for fixed routing is shown in Table 31, varying only in the capability of machines 2 and 3. Components 1 and 3 are always directed to machine 2 and components 2 and 4 are always directed to machine 3. The volume of each component type and relative processing times in this data set allow the workload to be balanced between machines 2 and 3.

In a similar manner to the previous series, flowtime and machine use results are shown in Tables 32 and 33. It may be seen that fixed routing has indeed reduced the setting time on machines 2 and 3 as expected but the flow time is greater for each component type. From tables of selected comparisons (Tables 34 and 35), this difference is significant. The "first free" machine rule always handles consecutive jobs, which arrive rather close to one another, more efficiently than fixed routing and hence the overall work in progress level in this area and flow times will be lower.

Tables 34 and 35 present comparisons of results for these rules with the best results, i.e. using the earliest expected start rule, from both series 1 dispersed buffers and series 2 shared buffers. Flowtime results for the earliest expected start rule on dispersed buffers and shared buffers are no worse and no better than results using the "first free" machine rule. Compared with fixed routing, the earliest expected start rule is better for all components when using shared allocated buffers and better for components 2 and 4 only (i.e. those components which went through machine 3) when using dispersed buffers. It is clear that

Table 31 Operation data used for series 3

Component	Op. no.	Op. time and capable machines 1	2	3	4
1	1	1.00			
	2		1.25		
	3				1.00
2	1	1.50			
	2			1.88	
	3				1.50
3	1	1.80			
	2		2.25		
	3				1.80
4	1	2.50			
	2			3.12	
	3				2.50

Other conditions:

Set-up times:	30 minutes
Inter-operational transport time:	30 minutes
Batch size:	50
Mean time to repair:	100 minutes
Mean time between failures:	13.2 hours

Notes:

Outer machines are bottlenecks

Expected bottleneck machine utilisation is 65%

Table 32 Series 3: First free machine and fixed routing

Flowtime results

Rule	Measure	Flowtime			
		Comp 1	Comp 2	Comp 3	Comp 4
First free machine	Mean	1037	1178	1331	1423
	St. Dev.	24	27	32	27
	U. C. L.	1091	1240	1404	1483
	L. C. L.	983	1116	1258	1363
Fixed routing	Mean	1054	1214	1360	1460
	St. Dev.	26	30	34	30
	U. C. L.	1113	1283	1436	1527
	L. C. L.	994	1145	1283	1393

Table 33 Series 3: First free machine and fixed routing

Machine use results

Rule	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
First free machine	Mean	46.6	10.1	34.3	7.0
	St. Dev.	0.2	0.1	0.2	0.1
	U. C. L.	46.9	10.2	34.6	7.1
	L. C. L.	46.2	9.9	33.9	6.9
Fixed routing	Mean	40.8	5.6	40.5	4.4
	St. Dev.	0.2	0.0	0.2	0.1
	U. C. L.	41.3	5.7	41.1	4.5
	L. C. L.	40.2	5.5	40.0	4.3

Table 34 Series 3: First free machine and fixed routing

Comparisons of selected flowtime results

where a = first free machine rule
b = fixed routing
c = earliest expected start rule - shared buffers
d = earliest expected start rule - dispersed buffers

Rules	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
b - a	Mean	16.7	36.3	29.1	37.0
	St. Dev.	7.4	6.6	7.9	8.8
	U.C.L.	33.5	51.2	47.0	56.9
	L.C.L.	-0.1	21.4	11.2	17.1
d - a	Mean	7.6	9.8	16.9	14.0
	St. Dev.	6.1	7.4	8.1	9.0
	U.C.L.	21.4	26.6	35.3	34.4
	L.C.L.	-6.2	-7.0	-1.5	-6.4
c - a	Mean	4.7	2.2	14.1	7.7
	St. Dev.	4.7	6.5	7.4	6.4
	U.C.L.	15.3	17.0	30.7	22.3
	L.C.L.	-5.9	-12.6	-2.5	-6.9
d - b	Mean	-9.1	-26.5	-12.2	-23.0
	St. Dev.	6.2	7.1	7.1	8.1
	U.C.L.	4.9	-10.5	3.9	-4.7
	L.C.L.	-23.1	-42.5	-28.3	-41.3
c - b	Mean	-12.0	-34.1	-15.0	-29.3
	St. Dev.	5.1	5.9	5.9	5.9
	U.C.L.	-0.4	-20.8	-1.6	-15.8
	L.C.L.	-23.6	-47.4	-28.4	-42.8

Table 35. Series 3: First free machine and fixed routing

Comparison of selected machine use results

where a = first free machine rule
b = fixed routing
c = earliest expected start rule - shared buffers
d = earliest expected start rule - dispersed buffers

Rules	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
b - a	Mean	-5.8	-4.5	6.3	-2.6
	St. Dev.	0.2	0.0	0.2	0.1
	U.C.L.	-5.4	-4.4	6.7	-2.4
	L.C.L.	-6.3	-4.6	5.9	-2.8
d - a	Mean	-0.3	0.7	0.8	1.2
	St. Dev.	0.1	0.0	0.1	0.0
	U.C.L.	0.0	0.8	1.0	1.3
	L.C.L.	-0.6	0.6	0.5	1.2
c - a	Mean	2.5	1.4	-2.1	0.5
	St. Dev.	0.1	0.0	0.1	0.0
	U.C.L.	2.7	1.5	-1.9	0.6
	L.C.L.	2.4	1.3	-2.3	0.4
d - b	Mean	5.5	5.2	-5.5	3.8
	St. Dev.	0.2	0.0	0.2	0.1
	U.C.L.	6.0	5.3	-5.1	4.0
	L.C.L.	5.1	5.1	-6.0	3.7
c - b	Mean	8.4	5.9	-8.4	3.1
	St. Dev.	0.2	0.1	0.2	0.1
	U.C.L.	8.8	6.0	-7.9	3.3
	L.C.L.	7.9	5.7	-8.8	2.9

being able to take the state of the network into account when making a decision outweighs the gains in reduced setting time made by eliminating the decision.

Tables 36 and 37 show that the mean and standard deviation of maximum flow times observed for each decision rule change insignificantly with the decision rule for the rules tested.

Table 36 Series 3: First free machine and fixed routing

Maximum flowtimes

Rule	Measure	Maximum flowtime			
		Comp 1	Comp 2	Comp 3	Comp 4
First free machine	Mean	3370	3516	3611	3670
	St. Dev.	156	175	170	141
	U. C. L.	3723	3911	3997	3988
	L. C. L.	3017	3121	3226	3352
Fixed routing	Mean	3403	3557	3620	3707
	St. Dev.	156	174	167	163
	U. C. L.	3755	3951	3998	4075
	L. C. L.	3052	3163	3242	3338

Table 37 Series 3: First free machine and fixed routing

Comparison of selected maximum flowtimes

where: a = first free machine rule

b = fixed routing

c = earliest expected start rule - shared buffers

d = earliest expected start rule - dispersed buffers

Rules	Measure	Maximum flowtimes Comp 1	Comp 2	Comp 3	Comp 4
b - a	Mean	34	40	9	37
	St. Dev.	39	65	52	62
	U. C. L.	123	187	127	177
	L. C. L.	-56	-106	-110	-104
d - a	Mean	111	12	56	57
	St. Dev.	56	38	31	43
	U. C. L.	236	99	127	155
	L. C. L.	-15	-74	-15	-40
c - a	Mean	128	3	92	45
	St. Dev.	67	41	54	58
	U. C. L.	280	97	215	176
	L. C. L.	-23	-90	-31	-87
d - b	Mean	77	-28	47	21
	St. Dev.	63	65	51	33
	U. C. L.	219	120	163	95
	L. C. L.	-65	-176	-68	-54
c - b	Mean	95	-37	83	8
	St. Dev.	70	56	72	68
	U. C. L.	254	91	246	163
	L. C. L.	-65	-165	-80	-147

8.6 Series 4 - Centre machines as bottlenecks

The basic data set was altered to make the 2 alternate machines 2 and 3 the rate limiting or bottleneck processors. For each component the second operation time was halved and became the new operation time for operations 1 and 3. The old times for operations 1 and 3 were doubled to become the new time for operation 2. Thus the total processing time per batch and component remain the same and average maximum machine loading is still 65%. All other conditions were similar and the new operational data set is summarised in table 38.

Ten replications were made using each rule in a similar manner to all previous series. Component flowtime and machine use results are shown in Tables 39 and 40.

Reference to Tables 39 and 40 indicates immediately that the random rule flowtime results are slightly lower under the new loading pattern than in earlier series and that flowtimes using the other rules are considerably lower than in earlier series. It is clarified here that the dispersed buffer arrangement is being used for Series 4, such that a routing decision is made on completion of processing on machine 1 and before inter-operational transit.

Tables 41 and 42 show how the performance of each rule compares with the previously used rule. The ratio rule makes a significant improvement to flowtime when compared with random allocation. Considering the number of jobs in the queue makes a further improvement to flowtime but a more accurate examination of the workload has barely any effect. However, the earliest expected start rule makes some improvement for 3 of the components over rule 4 and rule 3.

Table 38 Operation data used for series 4

Component	Op. no.	Op. time and capable machines 1	2	3	4
1	1	0.625			
	2		2.00	2.00	
	3				0.625
2	1	0.94			
	2		3.00	3.00	
	3				0.94
3	1	1.125			
	2		3.60	3.60	
	3				1.125
4	1	1.56			
	2		5.00	5.00	
	3				1.56

Other conditions:

Set-up times:	30 minutes
Inter-operational transport time:	30 minutes
Batch size:	50
Mean time to repair:	100 minutes
Mean time between failures:	13.2 hours

Notes:

INNER machines are bottlenecks

Expected bottleneck machine utilisation is 65%

Table 39 Series 4: Centre machines are bottleneck

Flowtime results

Rule	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	934	1050	1130	1231
	St. Dev.	21	29	22	22
	U. C. L.	983	1116	1179	1281
	L. C. L.	886	985	1081	1181
2. Ratio	Mean	749	851	933	1013
	St. Dev.	14	16	16	15
	U. C. L.	780	888	969	1047
	L. C. L.	719	815	897	978
3. No. jobs	Mean	685	793	882	970
	St. Dev.	10	12	14	13
	U. C. L.	708	821	912	998
	L. C. L.	662	765	851	942
4. LWINQ	Mean	674	784	877	966
	St. Dev.	9	11	11	9
	U. C. L.	695	807	903	987
	L. C. L.	653	760	852	945
5. EES	Mean	649	759	847	951
	St. Dev.	13	15	17	17
	U. C. L.	679	793	886	988
	L. C. L.	620	725	809	914

Table 40 Series 4: Centre machines are bottleneck

Machine usage results

Rule	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
1. Random	Mean	64.6	8.9	65.5	9.1
	St. Dev.	0.4	0.1	0.4	0.1
	U.C.L.	65.5	9.0	66.5	9.2
	L.C.L.	63.6	8.7	64.5	8.9
2. Ratio	Mean	65.0	10.5	65.0	10.6
	St. Dev.	0.3	0.1	0.3	0.0
	U.C.L.	65.6	10.7	65.6	10.7
	L.C.L.	64.5	10.4	64.4	10.5
3. No. jobs	Mean	73.3	10.5	56.8	8.2
	St. Dev.	0.2	0.0	0.4	0.1
	U.C.L.	73.6	10.5	57.7	8.3
	L.C.L.	72.9	10.4	55.9	8.0
4. LWINQ	Mean	73.1	10.1	56.9	8.1
	St. Dev.	0.1	0.0	0.4	0.1
	U.C.L.	73.4	10.2	57.8	8.2
	L.C.L.	72.8	10.1	56.0	7.9
5. EBS	Mean	67.5	9.6	62.5	8.9
	St. Dev.	0.2	0.0	0.3	0.0
	U.C.L.	68.0	9.7	63.2	9.0
	L.C.L.	67.1	9.5	61.8	8.8

Table 41 Series 4: Centre machines as bottleneck

Comparison of selected flowtime results

Rules	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
2 - 1	Mean	-184.9	-199.5	-197.4	-218.8
	St. Dev.	12.3	19.5	12.5	16.1
	U. C. L.	-157.1	-155.5	-169.1	-182.3
	L. C. L.	-212.7	-243.5	-225.7	-255.3
3 - 2	Mean	-64.6	-57.6	-51.0	-42.6
	St. Dev.	7.5	7.8	6.9	8.5
	U. C. L.	-47.7	-39.9	-35.0	-23.3
	L. C. L.	-81.5	-75.3	-66.6	-61.9
4 - 3	Mean	-11.0	-9.4	-4.4	-4.2
	St. Dev.	2.6	5.3	5.1	5.2
	U. C. L.	-5.1	2.5	7.2	7.5
	L. C. L.	-16.9	-21.3	-16.0	-15.9
5 - 4	Mean	-24.4	-24.3	-30.0	-14.9
	St. Dev.	9.4	10.5	11.1	12.7
	U. C. L.	-3.2	-0.6	-4.8	13.8
	L. C. L.	-45.6	-48.0	-55.2	-43.6
5 - 3	Mean	-35.4	-33.7	-34.4	-19.1
	St. Dev.	7.5	5.9	8.2	9.5
	U. C. L.	-18.3	-20.3	-15.9	2.5
	L. C. L.	-52.5	-47.1	-52.9	-40.7

Table 42 Series 4: Centre machines as bottleneck

Comparison of selected machine use results

Rules	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
2 - 1	Mean	0.5	1.6	-0.5	1.5
	St. Dev.	0.3	0.0	0.3	0.1
	U.C.L.	1.2	1.8	0.3	1.6
	L.C.L.	-0.2	1.5	-1.2	1.4
3 - 2	Mean	8.2	-0.1	-8.2	-2.4
	St. Dev.	0.1	0.0	0.1	0.0
	U.C.L.	8.5	0.0	-7.9	-2.3
	L.C.L.	7.9	-0.2	-8.5	-2.5
4 - 3	Mean	-0.1	-0.3	0.1	-0.1
	St. Dev.	0.1	0.1	0.1	0.1
	U.C.L.	0.0	-0.2	0.3	0.0
	L.C.L.	-0.3	-0.4	0.0	-0.2
5 - 4	Mean	-5.6	-0.6	5.6	0.8
	St. Dev.	0.1	0.1	0.1	0.0
	U.C.L.	-5.3	-0.4	5.9	0.9
	L.C.L.	-5.9	-0.7	5.3	0.7
5 - 3	Mean	-5.7	-0.9	5.7	0.7
	St. Dev.	0.1	0.1	0.1	0.0
	U.C.L.	-5.5	-0.8	5.9	0.8
	L.C.L.	-5.9	-1.0	5.5	0.6

Tables 43 and 44 present the mean, standard deviation and 95% confidence limits of maximum flowtimes observed during each replication, by decision rule, and selected comparisons of these results. It may be seen that the mean and standard deviation of maximum flowtimes observed are significantly higher (see Table 44) for the random rule. The mean and standard deviation of maximum flowtimes observed between replications is approximately similar for all other rules.

Table 43 Series 4: Centre machines as bottlenecks

Maximum flowtimes

Rule	Measure	Maximum flowtimes Comp 1	Comp 2	Comp 3	Comp 4
1. Random	Mean	3876	3928	4009	4030
	St. Dev.	250	213	245	222
	U. C. L.	4441	4409	4563	4532
	L. C. L.	3311	3447	3456	3528
2. Ratio	Mean	2805	2841	2879	2937
	St. Dev.	128	131	132	119
	U. C. L.	3094	3136	3177	3207
	L. C. L.	2516	2545	2581	2667
3. No. jobs	Mean	2497	2608	2616	2721
	St. Dev.	100	71	88	106
	U. C. L.	2723	2768	2814	2962
	L. C. L.	2271	2448	2417	2481
4. LWINQ	Mean	2397	2481	2607	2707
	St. Dev.	129	109	99	86
	U. C. L.	2688	2727	2832	2902
	L. C. L.	2107	2235	2382	2512
5. EES	Mean	2310	2378	2534	2612
	St. Dev.	110	131	133	126
	U. C. L.	2558	2675	2835	2897
	L. C. L.	2062	2082	2232	2328

Table 44 Series 4: Centre machines as bottleneck

Comparison of selected maximum flowtimes

Rules	Measure	Maximum flowtime Comp 1	Comp 2	Comp 3	Comp 4
2 - 1	Mean	-1072	-1088	-1130	-1093
	St. Dev.	257	193	263	201
	U. C. L.	-490	-651	-535	-638
	L. C. L.	-1653	-1524	-1726	-1548
3 - 2	Mean	-307	-232	-263	-216
	St. Dev.	127	134	118	124
	U. C. L.	-21	71	3	66
	L. C. L.	-594	-536	-529	-497
4 - 3	Mean	-100	-127	-9	-15
	St. Dev.	74	67	58	78
	U. C. L.	68	25	122	162
	L. C. L.	-267	-280	-140	-191
5 - 4	Mean	-88	-103	-73	-95
	St. Dev.	129	121	130	133
	U. C. L.	205	171	222	206
	L. C. L.	-380	-376	-368	-396
5 - 3	Mean	-187	-230	-82	-109
	St. Dev.	117	132	143	140
	U. C. L.	77	69	243	206
	L. C. L.	-451	-529	-406	-425

8.7 Series 5 Non-identical machines

The data set used for series 4 was further altered to give the two alternate machines non-identical capabilities, but still leaving these two machines with the highest workload. All other conditions were similar and the data set is summarised in Table 45.

Ten replications of each rule to be tested were made in a similar manner to all previous series. It is not appropriate to use random allocation for non-identical machines since any major difference in capability will unfairly load, or even overload, one machine. The random rule was not used in this configuration.

The ratio rule also needed adjustment according to the capabilities of the machines to handle different components. Thus components are routed to machines 2 and 3 in the following ratios, according to the processing capabilities in table 45:

Component	Machine 2	Machine 3
1	0.75	0.25
2	0.56	0.33
3	0.50	0.50
4	0.60	0.40

Results from runs using each rule are presented in tables 46 and 47. The earliest expected start rule appears to yield the shortest flowtimes again.

Table 49 Operation data used for series 5

Component	Op. no.	Op. time and capable machines 1	2	3	4
1	1	0.63			
	2		1.00	3.00	
	3				0.63
2	1	0.94			
	2		2.00	4.00	
	3				0.94
3	1	1.125			
	2		3.60	3.60	
	3				1.125
4	1	1.56			
	2		4.00	6.00	
	3				1.56

Other conditions:

Set-up times:	30 minutes
Inter-operational transport time:	30 minutes
Batch size:	50
Mean time to repair:	100 minutes
Mean time between failures:	13.2 hours

Notes:

INNER machines are bottlenecks, and not necessarily identical
Expected bottleneck machine utilisation is 65%

Table 46 Series 5: Non-identical machines

Flowtime results

Rule	Measure	Flowtime Comp 1	Comp 2	Comp 3	Comp 4
2. Ratio	Mean	612	717	816	893
	St. Dev.	6	6	7	6
	U. C. L.	626	731	833	908
	L. C. L.	599	703	800	878
3. No. jobs	Mean	609	712	807	878
	St. Dev.	7	9	8	8
	U. C. L.	626	731	825	896
	L. C. L.	592	693	789	860
4. LWINQ	Mean	595	700	801	868
	St. Dev.	7	8	7	6
	U. C. L.	610	717	817	881
	L. C. L.	580	683	785	855
5. EES	Mean	568	676	765	850
	St. Dev.	5	7	7	6
	U. C. L.	580	691	781	863
	L. C. L.	556	661	749	837

Table 47 Series 5: Non-identical machines

Machine use results

Rule	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
2. Ratio	Mean	58.6	11.7	58.3	7.9
	St. Dev.	0.2	0.1	0.2	0.0
	U.C.L.	59.1	11.9	58.8	8.0
	L.C.L.	58.0	11.6	57.8	7.8
3. No. jobs	Mean	64.3	12.1	55.9	6.4
	St. Dev.	0.2	0.0	0.4	0.1
	U.C.L.	64.7	12.2	56.7	6.5
	L.C.L.	63.8	12.0	55.0	6.2
4. LWINQ	Mean	64.4	12.0	55.0	6.2
	St. Dev.	0.2	0.0	0.3	0.1
	U.C.L.	64.8	12.1	55.7	6.3
	L.C.L.	63.9	11.9	54.2	6.1
5. EES	Mean	60.2	11.2	60.9	7.0
	St. Dev.	0.3	0.1	0.2	0.0
	U.C.L.	60.8	11.5	61.5	7.1
	L.C.L.	59.6	11.2	60.4	6.9

From Table 47, the ratio rule has been able to balance workload on machines 2 and 3, but not necessarily setting time, now that machine processing times are different. The earliest expected start rule has also balanced the workload, although the "least work in next queue" rule and "number of jobs" rule still use machine 2 more than machine 3.

From Tables 48 and 49 of selected rule comparisons, the "number of jobs" rule makes a significant improvement to flowtime of components 3 and 4 only, in comparison with the ratio rule. The workload rule however reduced the flowtime of all components in comparison with the ratio rule, and the earliest expected start rule makes a further significant reduction to flow time of all components (Table 48).

Although the earliest expected start rule has balanced workload across machines 2 and 3, the total machine working time is greater on both machines under this rule and the setting times are both lower. It is recognised that the earliest expected start decision rule does not take current machine set ups into account. It is observed that the ratio rule incurs higher total setting up time than all the other rules. Reference to all previous results finds this to be true in all cases examined.

The "number of jobs in the next queue" rule is the only rule used here which does not directly take the different machine processing speeds into account. However, this rule will not overload machines at this level of work-in-progress because the faster machine should pull jobs out of the queue faster and hence attract further jobs into the queue. It is also noted that the faster machine is designated route 1, and hence ties will be broken onto the fastest machine. A situation where ties are

Table 48 Series 5: Non-identical machines

Comparison of selected flowtime results

Rules	Measure	Flowtime			
		Comp 1	Comp 2	Comp 3	Comp 4
3 - 2	Mean	-3.5	-4.9	-9.6	-15.3
	St.Dev.	2.6	3.7	4.1	4.0
	U.C.L.	2.3	3.5	-0.2	-6.2
	L.C.L.	-9.3	-13.3	-19.0	-24.4
4 - 2	Mean	-17.9	-16.8	-15.4	-24.9
	St.Dev.	2.3	3.5	3.5	5.0
	U.C.L.	-12.7	-9.0	-7.4	-13.6
	L.C.L.	-23.1	-24.6	-23.4	-36.2
5 - 2	Mean	-44.3	-41.0	-51.5	-42.9
	St.Dev.	2.3	4.0	5.2	6.3
	U.C.L.	-39.2	-32.0	-39.6	-28.5
	L.C.L.	-49.4	-50.0	-63.4	-57.3
5 - 4	Mean	-26.4	-24.2	-36.1	-18.0
	St.Dev.	2.8	3.8	2.9	3.3
	U.C.L.	-20.0	-15.7	-29.6	-10.6
	L.C.L.	-32.8	-32.7	-42.6	-25.4

Table 49 Series 5: Non-identical machines

Comparison of selected machine use results

Rules	Measure	M/c 2 Working	M/c 2 Setting	M/c 3 Working	M/c 3 Setting
3 - 2	Mean	5.7	0.4	-2.5	-1.5
	St. Dev.	0.1	0.1	0.2	0.0
	U.C.L.	5.9	0.5	-2.0	-1.4
	L.C.L.	5.5	0.3	-2.9	-1.6
4 - 2	Mean	5.8	0.3	-3.4	-1.7
	St. Dev.	0.1	0.0	0.1	0.0
	U.C.L.	6.1	0.4	-3.0	-1.6
	L.C.L.	5.5	0.2	-3.7	-1.8
5 - 2	Mean	1.7	-0.4	2.6	-0.9
	St. Dev.	0.1	0.0	0.1	0.0
	U.C.L.	1.8	-0.3	2.8	-0.8
	L.C.L.	1.5	-0.5	2.4	-1.0
5 - 4	Mean	-4.1	-0.6	6.0	0.8
	St. Dev.	0.1	0.0	0.2	0.0
	U.C.L.	-3.8	-0.6	6.4	0.9
	L.C.L.	-4.4	-0.7	5.5	0.7

broken onto a slower machine should clearly be avoided, unless there is some quality or cost advantage to the slower alternate machine.

The maximum flow times are observed to change only slightly between decision rules (Tables 50 and 51). The maximum flowtimes observed for component 1 are decreased by using the LWINQ or EES rule in comparison with the ratio rule but there is little effect on other components.

Table 50 Series 5: Non-identical machines

Maximum flowtime

Rule	Measure	Maximum flowtime Comp 1	Comp 2	Comp 3	Comp 4
2. Ratio	Mean	2196	2297	2455	2433
	St. Dev.	63	66	84	92
	U. C. L.	2339	2445	2644	2641
	L. C. L.	2054	2149	2265	2226
3. No. jobs	Mean	2300	2302	2425	2545
	St. Dev.	106	80	82	85
	U. C. L.	2541	2482	2610	2737
	L. C. L.	2059	2122	2240	2354
4. LWINQ	Mean	1999	2172	2292	2331
	St. Dev.	36	79	59	62
	U. C. L.	2079	2350	2424	2470
	L. C. L.	1918	1993	2159	2192
5. EES	Mean	1942	2022	2221	2225
	St. Dev.	49	39	101	70
	U. C. L.	2054	2110	2449	2384
	L. C. L.	1831	1935	1994	2066

Table 51 Series 5: Non-identical machines

Comparison of selected maximum flowtimes

Rules	Measures	Maximum flowtimes			
		Comp 1	Comp 2	Comp 3	Comp 4
3 - 2	Mean	104	5	-30	112
	St. Dev.	129	92	96	119
	U. C. L.	395	212	188	382
	L. C. L.	-188	-202	-247	-157
4 - 2	Mean	-198	-126	-163	-102
	St. Dev.	62	98	98	118
	U. C. L.	-57	95	59	165
	L. C. L.	-339	-346	-386	-369
5 - 2	Mean	-254	-275	-233	-209
	St. Dev.	98	81	135	126
	U. C. L.	-32	-92	71	76
	L. C. L.	-476	-458	-538	-493
5 - 4	Mean	-56	-149	-70	-107
	St. Dev.	49	106	91	102
	U. C. L.	55	91	136	124
	L. C. L.	-168	-389	-276	-337

8.8 General conclusions

Alternative routes comprising single operations only have been investigated in order to make a more informed approach to managing alternative routes comprising more operations.

As is the case with despatching rules, it is immediately apparent that different rules are appropriate for different objectives. Table 52 summarises the performance of the decision rules when the alternative operations do not constitute the bottleneck, against three objectives:

(i) balance workload across the machines

(ii) minimise flow time

(iii) minimise total setting time

Although the "first free" machine rule can only be used with pooled buffers, it has been compared with the other rules on shared buffers. The fixed routing experiments were carried out on dispersed buffers and hence these results are included for comparison.

The ratio and random rules achieve workload balance directly. The other rules do not achieve workload balance when used alone but if a second rule were used to break ties, it is considered that workload balance could be improved. The ratio rule as defined here should not be used to achieve workload balance where there is a regular arrival pattern, e.g. slow job, fast job, which could coincide with the routing pattern. In this case, the probalistic traffic splitting (Section 3.10) would

Table 52 Suitability of alternate routing rules to different objectives for single operation alternatives which are not rate limiting

Rating of 1 = best

SHARED BUFFERS (identical machines)

Rules	Workload balance	Flowtime	Min. setting time
Random	1	2	1
Ratio	1	2	2
Least no. jobs	2	2	1
Least workload	2	2	1
EES	2	1	2
First free m/c	2	1	1

DISPERSED BUFFERS (identical machines)

Rules	Workload balance	Flowtime	Min. setting time
Random	1	2	2
Ratio	1	1	4
Least no. jobs	2	2	2
Least workload	2	2	2
EES	2	1	3
Fixed routing	1/2	1	1

be more appropriate (equivalent to the random rule but with unequal chances of choosing each route).

The earliest expected start (EES) rule yields good flowtime results under shared and dispersed buffer configurations, matching the flowtimes achieved by the first free machine rule for a pooled buffer and the fixed routing rule to dispersed buffers. The surprising result is the flowtime achieved by the ratio rule for dispersed buffers, particularly considering the lack of distinction for this rule on shared buffers.

There is a clear pattern for total set up time. The ratio rule performs poorly. Fixed routing is clearly best, but the random rule, least number of jobs or least workload in the next queue perform reasonably. The EES rule performs worse than average against this criterion.

Table 53 summarises the results where the single operation alternatives are the bottleneck in the system. The random and ratio rules achieve workload balance (where used). The random rule could have been used on non-identical machines using probabilistic traffic splitting, taking the ratios used for the ratio rule as the probabilities for each alternative. The interesting result is the good achievement of workload balance by the EES rule on non-identical machines.

The EES rule achieves the shortest flow times. The differences in flowtime performance are more marked when the centre machines are bottlenecks. The random rule performed worst and the ratio rule results were also very poor.

Table 53 Suitability of alternate routing rules to different objectives for single operation alternatives where the alternatives are rate limiting

Rating of 1 = best

IDENTICAL MACHINES (dispersed buffers)

Rules	Workload balance	Flowtime	Min. setting time
Random	1	4	1
Ratio	1	3	2
Least no. jobs	3	2	1
Least workload	3	2	1
EES	2	1	1

NON IDENTICAL MACHINES (dispersed buffers)

Rules	Workload balance	Flowtime	Min. setting time
Ratio	1	3	2
Least no. jobs	2	3	1
Least workload	2	2	1
EES	1	1	1

There was less difference in performance of the rules against the total setting time criterion. The ratio rule performed poorly compared with all the others.

9. Partial route alternatives

The phrase "partial route" alternative is used here to describe an alternate route of one or more operations taking place at one or more machines. The decision rules for partial routes must take into account any differences in processing speed among the machines on the partial route. Identification of the bottleneck or rate limiting machine will prevent overloading the partial route. For example, the "first free" machine and "least work in next queue" rules will not work unless the first machine on the partial route is the bottleneck, or unless inventory at that operation is strictly limited.

9.1 Machine configuration and data

The machine configuration used to test decision rules for partial routes is shown in Figure 34. The data set used for previous series was further amended to create three operations for each component on the alternative route (table 54). It will be observed from Table 54 that for components 1 and 3 the second operation on the partial route has the longest processing time. However the longest processing time on the alternative route of components 2 and 4 is at the third operation.

In this data set, the centre machines (2, 3, 4 and 5) are, in general, more heavily loaded than the outside machines (1 and 6). All other conditions are similar to previous series.

FIG. 34 MACHINE CONFIGURATION USED TO INVESTIGATE PARTIAL ROUTES

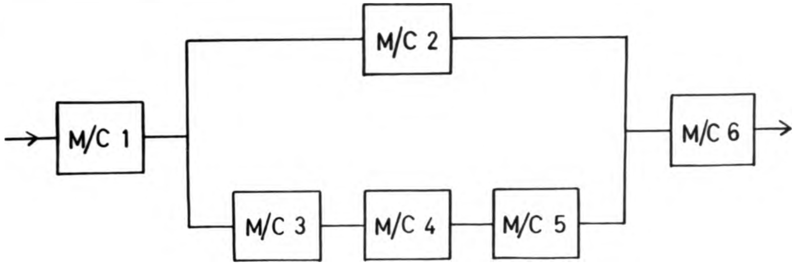


Table 54 Operation data used for series 6

Comp	Op. No.	Op. time and capable machines 1	2	3	4	5	6
1	1	0.63					
	2		1.0				
	ALT2			0.7	+ 3.0	+ 1.5	
	3						0.63
2	1	0.94					
	2		2.0				
	ALT2			1.1	+ 2.3	+ 4.0	
	3						0.94
3	1	1.13					
	2		3.6				
	ALT2			1.6	+ 3.6	+ 1.0	
	3						1.13
4	1	1.56					
	2		4.0				
	ALT2			0.5	+ 1.5	+ 2.0	
	3						1.56

Other conditions:

Set-up times: 30 minutes

Inter-operational transport time: 30 minutes

Batch size: 50

Mean time to repair: 100 minutes

Mean time between failures 13.2 hours

Notes:

Machines 2, 3, 4 and 5 are more heavily loaded than 1 and 6

Expected maximum machine utilisation is 65%

9.2 Description of rules

A "ratio" rule (rule 1) similar to the rule used to investigate single operation alternatives, was used as a comparator against which other rules are measured.

Rule 2 considers the number of jobs in the whole of each partial route. In this case, one partial route comprises machine 2 only (route 1), and the other partial route comprises machines 3, 4 and 5 (route 2).

Rules 3, 4 and 5 identify the bottleneck machine along the partial route, according to the planned or expected workload for that machine, and consider the state of the partial route from the decision point up to and including the bottleneck.

9.2.1 "Ratio"

Similar to the rule used for single operation alternatives, jobs are directed to each alternate route to maintain some predetermined ratio. The ratio is in inverse proportion to the ratio of the longest processing time in each alternate route for that component. The ratios for the data set in Table 54 are therefore:

Component	Processing time ratio route 1 - route 2	Decision rule ratio
1	1.00 : 3.00	3 : 1
2	2.00 : 4.00	2 : 1
3	3.60 : 3.60	1 : 1
4	4.00 : 2.00	1 : 2

The purpose of the ratio is to maintain some sort of workload balance between the routes, according to their different capabilities, and to prevent overloading of one partial route although work would still be directed to a route which could currently be suffering a severe breakdown.

9.2.2 Least number of jobs in the partial route (LNJIR)

Each operation on the partial route is considered in turn. The workload already waiting for the machine or machines, capable of doing the operation is counted, plus jobs already in transit which have been directed to this machine, plus any job currently being processed. (If more than one machine is capable of the operation, the total number of jobs awaiting this operation is divided by the number of capable machines, to yield a job portion for the machine). The number of jobs found at each operational stage is summed to produce a quantity of jobs observed on that partial route.

The decision job is directed to the partial route which has the least number of jobs.

9.2.3 Least number of jobs yet to be processed by bottleneck machines (LNJBB)

The first stage for rules 3, 4 and 5 is to identify the bottleneck. The bottleneck machine on each partial route is identified here by examining the total potential workload for that machine if all jobs went along that partial route. The use of dynamic routing rules based on the state of the network precludes knowledge of the proportion of jobs that will take each partial route. Since jobs of any one component type will load the machines on the partial route in the same proportions regardless of the demand volume of the component it should be a fair assessment. However, different components will load machines differently and different proportions of each component may choose a particular partial route. Hence this bottleneck assessment can only be a rough guide.

Having found the bottleneck, the number of jobs observed to be waiting, in transit or being processed, is counted from the decision point up to and including any jobs being processed on the bottleneck machine, but excluding any jobs which have been completed by the bottleneck machine.

9.2.4 Workload expected by the bottleneck machine (LWKBB)

The list of jobs found by the method in 9.2.3 is converted into workload for the bottleneck machine. This represents the amount of work currently before the bottleneck machine which must be processed before the decision job, assuming a first come, first served queue discipline.

9.2.5 Earliest expected start on the bottleneck machine (EESB)

The workload information in 9.2.4 is enhanced by knowledge about the status of the remaining processing time for any current jobs on the bottleneck machine and the duration of any breakdown on this machine, in a similar manner to the earliest expected start rule used in previous series.

9.3 Series 6 Partial routes, centre machines as bottlenecks

Ten replications were made for each decision rule under exactly the same running conditions used for all previous series (five warm up periods, results collected from next 50 periods etc.). This series of experiments aimed to assess the relative performance of the basic rules on partial routes.

9.3.1 Machine use

Table 55 contains the proportions of time spent working and being set up by the inner machines 2, 3, 4 and 5. Machine 4 is most used and is the bottleneck in the partial route. It is observed that the setting time proportions are identical through the alternate route because all the set up times are identical in the data set and the FCFS rule forces the same component sequence through each machine.

The ratio rule does not appear to have achieved workload balance because the bottleneck operation times which were used to create the ratio for each component occur at different machines. An overall ratio taking into account total demand by component and therefore the potential workload on each machine can be used to achieve workload balance and is investigated in series 8.

The earliest expected start rule, which achieved workload balance over non-identical machines for single operation alternatives in series 5 did not achieve the same effect in series 6. In contrast to the ratio rule which had previously used a separate ratio for each component, the earliest expected start rule already considers the overall bottleneck in the partial route rather than the rate-limiting operation for a particular component. The balanced result of series 5 is

Table 55 Series 6: Partial routes, centre machines as bottlenecks - Machine use results

	Machine no. 2	3	4	5
1. Ratio rule				
Mean % working	51.6	16.2	43.2	33.6
Mean % setting	10.6	8.6	8.6	8.6
2. Least number of jobs in partial route				
Mean % working	63.7	14.1	40.5	31.2
Mean % setting	12.0	6.5	6.5	6.5
3. Least number of jobs before bottleneck				
Mean % working	60.4	15.6	44.7	34.5
Mean % setting	11.4	7.1	7.1	7.1
4. Least workload before bottleneck				
Mean % working	57.1	16.7	47.1	36.8
Mean % setting	10.7	7.4	7.4	7.4
5. Earliest expected start at bottleneck				
Mean % working	58.5	16.1	45.1	35.5
Mean % setting	11.0	7.2	7.2	7.2

particularly interesting and provoked thought on how workload balance could be achieved for one of the adaptive rules. This idea is developed and tested in series 8.

9.3.2 Output

Table 56 shows the number of batches directed to each alternative under each rule. Because the input to machine 1 is controlled by the same mean arrival rate for each replication and each rule, and results are averaged from replications of the same random number stream sets, i.e. common random numbers, the overall number of batches of each component to pass through the system under each rule is almost identical. Therefore the total number of batches is listed once. However, it will be observed that not all the route 1 plus route 2 results match these overall values exactly because of integer rounding errors and different treatment under different decision rules.

In Table 56, it is observed that the ratio rule has strictly maintained the proportion of batches through route 1 according to the ratios specified. The other rules do not take component type into account and the result is a similar division between routes for all components.

9.3.3 Flowtime

Table 57 contains the overall flowtime results by component for each rule. It is difficult to compare the ratio rule flowtimes with the other rules because overall flowtimes for components 1 and 2 are lower than all the other results and overall flowtimes for components 3 and 4 are higher than all other results for those

Table 56 Series 6: Partial routes, centre machines as bottlenecks - Output results (numbers of batches)

	Component no. 1	2	3	4
Overall	1645	1085	898	647

1. Ratio rule

	1	2	3	4
Route 1	1233	727	449	213
Route 2	411	358	449	433
% batches thro' route 1	75	67	50	33

2. Least number of jobs in partial route

	1	2	3	4
Route 1	1038	715	607	429
Route 2	606	370	292	217
% batches thro' route 1	63	66	68	66

3. Least number of jobs before bottleneck

	1	2	3	4
Route 1	987	665	570	417
Route 2	658	420	328	229
% batches thro' route 1	60	61	63	64

4. Least workload before bottleneck

	1	2	3	4
Route 1	995	629	530	388
Route 2	650	457	368	258
% batches thro' route 1	60	58	59	60

5. Earliest expected start at bottleneck

	1	2	3	4
Route 1	1033	640	543	395
Route 2	612	445	358	252
% batches thro' route 1	63	59	60	61

Table 57 Series 6: Partial routes, centre machines as bottlenecks - Summary of flowtime results

	Component no.			
	1	2	3	4
1. Ratio rule				
Mean flowtime	587	735	846	905
St. Dev. of observ'ns	359	418	433	449
2. Least number of jobs in partial route				
Mean flowtime	641	763	836	852
St. Dev. of observ'ns	366	425	428	426
3. Least number of jobs before bottleneck				
Mean flowtime	630	761	828	841
St. Dev. of observ'ns	374	432	439	429
4. Least workload before bottleneck				
Mean flowtime	614	768	830	838
St. Dev. of observ'ns	370	443	444	428
5. Earliest expected start at bottleneck				
Mean flowtime	599	751	814	828
St. Dev. of observ'ns	362	434	436	426

components. However, Table 59 compares the adaptive rules and the performance of the earliest expected start rule with the ratio rule.

It is observed that counting the number of jobs before the bottleneck improved the flowtime of three out of four components in preference to counting the number of jobs in the whole partial route. Assessing the work load before the bottleneck improved the flowtime for two out of four components compared with counting the number of jobs in the whole partial route, but all flowtimes improve significantly if that workload is modified by the current state of the bottleneck machine. This result confirms the importance of assessing the current state of the bottleneck machine and is investigated further in the next series.

Table 58 contains summaries of flowtimes for each route for each rule. Clearly the flowtimes in route 2 are significantly larger with an increase in spread of observed flowtimes. On route 1, the "least workload before the bottleneck" rule (LWKBB) and the "earliest start at the bottleneck" rule (EESB) perform better than the ratio rule for all components. However, trends in the results in route 2 are less clear. The ratio rule performs best for components 2 and 3 with regard to other rules, but poorly for components 1 and 4. The LWKBB rule performs poorly for all components compared with the ratio rule. The EESB rule performs well for component 1, worse for components 2 and 3 and matches the ratio rule for component 4. Tables 60 and 61 show selected flowtime comparisons indicating that the "least number of jobs in partial route" rule achieves the highest flowtimes through route 1 but the lowest flowtimes through route 2. This is a clear indication of the way in which route 1 is more heavily loaded because of the trio of machines in route 2.

Table 58 Series 6: Partial routes, centre machines as bottlenecks - **Summary of flowtime results by route**

Route 1

		Component no. 1	2	3	4
Ratio	Mean route 1	506	605	709	750
	St. Dev. route 1	320	349	376	386
LNJIR	Mean route 1	544	639	760	802
	St. Dev. route 1	335	366	402	412
LNJBB	Mean route 1	496	596	719	762
	St. Dev. route 1	311	348	386	391
LWKBB	Mean route 1	469	568	689	732
	St. Dev. route 1	293	329	370	374
EESB	Mean route 1	472	562	675	731
	St. Dev. route 1	287	325	365	372

Route 2

		1	2	3	4
Ratio	Mean route 2	831	1000	983	981
	St. Dev. route 2	360	421	442	459
LNJIR	Mean route 2	808	1002	994	950
	St. Dev. route 2	357	428	436	437
LNJBB	Mean route 2	832	1021	1016	982
	St. Dev. route 2	370	426	461	455
LWKBB	Mean route 2	838	1043	1033	997
	St. Dev. route 2	363	432	463	454
EESB	Mean route 2	815	1023	1028	981
	St. Dev. route 2	372	426	453	460

Table 59 Series 6: Partial routes, centre machines as bottleneck - Comparison of selected flowtimes

Rules	Measure	Comp 1	Comp 2	Comp 3	Comp 4
3 - 2	Mean	-11	-2	-8	-11
	St. Dev.	2	3	2	4
	U. C. L.	-6	4	-3	-2
	L. C. L.	-15	-8	-14	-21
4 - 2	Mean	-27	6	-6	-14
	St. Dev.	2	5	3	6
	U. C. L.	-22	16	0	-1
	L. C. L.	-32	-5	-12	-27
5 - 2	Mean	-42	-12	-22	-23
	St. Dev.	3	4	4	5
	U. C. L.	-35	-3	-12	-13
	L. C. L.	-48	-21	-31	-34
5 - 1	Mean	13	16	-31	-77
	St. Dev.	3	3	4	3
	U. C. L.	19	23	-23	-71
	L. C. L.	6	8	-40	-83

where:

Rule 1 = Ratio based on bottleneck operation times
Rule 2 = Least number of jobs in partial route
Rule 3 = Least number of jobs before bottleneck
Rule 4 = Least workload before bottleneck
Rule 5 = Earliest expected start at bottleneck

U. C. L. = Upper 95% confidence limit
L. C. L. = Lower 95% confidence limit

Table 60 Series 6: Partial routes, centre machines as bottleneck - Comparison of selected flowtimes - route 1

Rules	Measure	Comp 1	Comp 2	Comp 3	Comp 4
3 - 2	Mean	-48	-43	-41	-40
	St. Dev.	3	4	3	6
	U. C. L.	-42	-34	-33	-27
	L. C. L.	-55	-51	-49	-53
4 - 2	Mean	-75	-71	-72	-70
	St. Dev.	2	7	4	8
	U. C. L.	-70	-56	-64	-52
	L. C. L.	-80	-86	-80	-88
5 - 2	Mean	-72	-77	-86	-71
	St. Dev.	4	4	4	7
	U. C. L.	-64	-69	-78	-55
	L. C. L.	-80	-86	-94	-87
5 - 1	Mean	-34	-43	-34	-19
	St. Dev.	2	3	5	5
	U. C. L.	-29	-36	-24	-7
	L. C. L.	-39	-51	-44	-30

where:

Rule 1 = Ratio based on bottleneck operation times
Rule 2 = Least number of jobs in partial route
Rule 3 = Least number of jobs before bottleneck
Rule 4 = Least workload before bottleneck
Rule 5 = Earliest expected start at bottleneck

U. C. L. = Upper 95% confidence limit
L. C. L. = Lower 95% confidence limit

Table 61 Series 6: Partial routes, centre machines as bottleneck - Comparison of selected flowtimes - route 2

Rules	Measure	Comp 1	Comp 2	Comp 3	Comp 4
3 - 2	Mean	24	19	23	32
	St. Dev.	3	5	4	10
	U. C. L.	32	29	32	55
	L. C. L.	17	9	13	10
4 - 2	Mean	30	41	39	47
	St. Dev.	3	5	6	8
	U. C. L.	37	52	54	65
	L. C. L.	23	30	25	28
5 - 2	Mean	7	21	34	31
	St. Dev.	4	6	9	9
	U. C. L.	16	34	54	51
	L. C. L.	-2	7	14	11
5 - 1	Mean	-16	23	45	0
	St. Dev.	6	6	8	7
	U. C. L.	-3	37	64	15
	L. C. L.	-29	9	27	-16

where:

Rule 1 = Ratio based on bottleneck operation times
Rule 2 = Least number of jobs in partial route
Rule 3 = Least number of jobs before bottleneck
Rule 4 = Least workload before bottleneck
Rule 5 = Earliest expected start at bottleneck

U. C. L. = Upper 95% confidence limit
L. C. L. = Lower 95% confidence limit

9.4 Series 7 Partial routes, centre machines as bottlenecks - investigation of earliest expected start rule.

The earliest expected start rule has performed well overall among the related number of jobs and workload rules.

Two further sets of runs were carried out to find out if one of its two extra items of information, current job being processed or current breakdown, was more important than the other. In addition to the results of these runs, the number of decisions where these data items were taken into account was recorded from one run to obtain an idea of their relative importance. Out of a total of 4745 decisions using the earliest expected start rule:

- 338 decisions included current breakdown duration and remaining processing time on machine 2 (route 1)
- 258 decisions included current breakdown duration and remaining processing time on the route 2 bottleneck machine
- 2973 decisions included remaining processing time on machine 2 (route 1) i.e.when there were no breakdowns
- 1967 decisions included remaining processing time on the bottleneck machine on route 2 i.e. when there were no breakdowns.

The apparent importance of remaining processing time of any current job at the bottleneck machine is confirmed by the overall flowtime results in Table 62. A negligible improvement in flowtimes is observed when repair times alone are taken into account but a significant change occurs if just the remaining processing time is taken into account in the decision. There is little difference between

Table 62 Series 7: Partial routes, centre machines as bottlenecks - EES rule investigation

	Component 1	2	3	4
4. Least workload before bottleneck				
Mean flowtime	614	768	830	838
St. Dev. of observ'ns	370	443	444	428
7. Least workload plus repair time only				
Mean flowtime	608	763	826	836
St. Dev. of observ'ns	370	444	447	433
6. Least workload plus remaining processing time only				
Mean flowtime	598	750	816	825
St. Dev. of observ'ns	359	430	436	422
5. Earliest expected start at bottleneck (i.e. including repair time and remaining processing time)				
Mean flowtime	599	751	814	828
St. Dev. of observ'ns	362	434	436	426

including remaining processing time only and including both items of information.

This result is important since repair times are inherently more difficult to estimate. Clearly the generality of this result depends on the level of breakdowns, the repair time pattern and any difference in breakdown patterns between the machines.

9.5 Series 8 Partial routes, inner machines are bottlenecks - Investigation of workload balance.

Workload balance was observed to be important in early alternative routing work, although Stecke and Solberg (52) obtained contradictory results. Workload balance was not achieved in series 6 and is investigated further here.

The calculation for the overall workload balance ratio proposed in section 9.3 is shown in Table 63. Ten replications using this ratio for all components were made under the usual conditions and the results are summarised in Tables 64 and 65. It can be seen that working time has been balanced across the bottleneck machines in each partial route.

Having achieved workload balance on partial routes for the non-adaptive ratio rule, it was decided to attempt workload balance for an adaptive rule. The "earliest expected start at the bottleneck" rule had performed well generally. The difference in use of the bottleneck machines on each route was assumed to be due to the way in which ties were broken, following machine 2 as the "first found" machine in the case of a tie. Using the ratio rule to break ties equally between the two routes did not balance the working time of machines 2 and 4 which were 57.0% and 47.4% respectively. The ratio was changed to the overall loading ratio calculated in Table 63. There was little change to the results. It was realised that the earliest expected start rule counts work from the decision point up to and including the bottleneck, i.e. work for machines 3 and 4, and therefore had an increased probability of reporting a higher waiting workload in contrast to only one machine on route 1. This is clearly not a fair comparison.

Table 63 Calculation of workload balance ratio

Stage 1: Potential number of arrivals of each component per 10,000 minute period:

Dividing 10,000 period duration by inter-arrival time:

Component 1:	10,000 + 307 = 32.6
Component 2:	10,000 + 462 = 21.6
Component 3:	10,000 + 554 = 18.1
Component 4:	10,000 + 769 = 13.0

Stage 2: Calculate potential workload on each machine

Workload ∝ No. arrivals x component operation time

For example, machine 2:

Component 1:	32.6 x 1.0	=	32.6
Component 2:	21.6 x 2.0	=	43.2
Component 3:	18.1 x 3.6	=	65.2
Component 4:	13.0 x 4.0	=	52.0
	Total		193.0

Similar calculations for other machines, yields:

Machine 3 total	82.1
Machine 4 total	232.2
Machine 5 total	179.4

Stage 3: Compare maximums on each route

(Max. route 1 + Max. route 2) = 193 / (193 + 232)
= 0.454

Route 1 can therefore process faster. 45.4% of work should be directed to route 2, and 54.6% should be directed to route 1.

Table 64 Series 8: Partial routes, inner machines are bottleneck - Investigation of workload balance

Summary of machine use	Machine no. 1	2	4	5
a Ratio rule (0.546/0.454)				
Mean % working	37.9	52.6	52.8	40.8
Mean % setting	12.1	11.0	9.6	9.6
b Workload at bottleneck machine				
Mean % working	37.9	55.1	52.4	40.6
Mean % setting	12.1	10.1	7.7	7.7
c Workload at bottleneck machine (and tie break with ratio)				
Mean % working	37.9	53.4	54.3	42.3
Mean % setting	12.1	9.7	7.9	7.9

Summary of output	Component 1	2	3	4
Overall	1645	1085	898	646
a Ratio rule (0.546/0.454)				
Batches through route 1	898	592	491	353
% batches thro' route 1	54.6	54.6	54.7	54.6
b Workload at bottleneck machine				
Batches through route 1	844	589	532	394
% batches thro' route 1	51.3	54.3	59.2	61.0
c Workload at bottleneck machine (and tie break with ratio)				
Batches through route 1	816	568	523	376
% batches thro' route 1	49.6	52.4	58.2	58.2

Table 66 Series 8: Partial route, inner machines are bottleneck - Investigation of workload balance

Summary of flowtimes	Component			
	1	2	3	4
a. Ratio rule (0.546,0.454)				
Overall flowtime	686	824	874	901
Overall st.dev.	405	458	456	457
Flowtime route 1	529	636	745	797
Route 1 st.dev.	333	376	401	412
Flowtime route 2	876	1052	1029	1027
Route 2 st.dev.	402	444	469	476
b. Workload at bottleneck machine				
Overall flowtime	669	799	840	849
Overall st.dev.	423	476	458	448
Flowtime route 1	438	553	685	729
Route 1 st.dev.	279	324	371	380
Flowtime route 2	912	1091	1067	1035
Route 2 st.dev.	411	463	478	480
c. Workload at bottleneck machine (and tie break with ratio)				
Overall flowtime	679	810	846	860
Overall st.dev.	424	472	455	447
Flowtime route 1	444	564	699	749
Route 1 st.dev.	277	328	371	387
Flowtime route 2	911	1081	1051	1014
Route 2 st.dev.	415	456	481	478

The program was altered to consider work waiting for the bottleneck machine only, without using a second rule to break ties. After initial favourable results, the usual ten replications were made. Workload was still not wholly balanced between the two bottleneck machines (see Table 64) and the ratio rule using a 50% split between routes was used to break ties. Ten replications were made . The results of these replications are summarised in Tables 64 and 65.

It appears that assessing the workload at the bottleneck machine and breaking ties has approached workload balance but has not achieved balance as well as the ratio rule. The workload balance ratio (0.546:0.464) should perhaps have been used instead, or there may be other factors which have not been identified yet.

In the output summary, the desired ratio has clearly been achieved by the ratio rule. The other rules show a wide range of proportions of jobs directed to route 1.

Table 65 contains the flowtime results, showing the increase in flowtime for the adaptive rule by adding a tie break. There appears to be a flowtime penalty for the approach to workload balance. This penalty is confirmed by the comparisons in Table 66 which shows penalties incurred by both rules.

Table 66 Series 8: Partial routes, inner machines are bottleneck - Investigation of workload balance

Comparison of selected flowtimes

	Component			
	1	2	3	4

a. Workload balance ratio minus processing time ratio

	1	2	3	4
Mean	99	89	29	-4
St. dev.	4	3	5	5
U. C. L.	107	96	39	7
L. C. L.	91	83	18	-14

b. Workload at bottleneck machine (and tie break with ratio) minus earliest expected start at bottleneck machine

	1	2	3	4
Mean	79	60	32	31
St. Dev.	3	5	6	4
U. C. L.	86	71	45	41
L. C. L.	73	48	19	22

9.6 Series 9 Partial routes, centre machines are bottlenecks - Feedback method

Consideration of the backward learning technique for data communications networks (section 3.10) gave rise to a feedback method for deciding which route to follow according to flowtimes recorded on each route.

Average flowtimes observed on each route were recorded. At the time of the decision, the route displaying the shortest flowtime was chosen. An example of the results from one replication are shown in Table 67. The flowtimes are indeed similar between the two routes, in contrast to Table 58 for example. However, all the flowtimes are high, and also highly variable. Two further replications were made whose results were not dissimilar to replication A in Table 67 (apart from the number of batches of component 4 on route 2). The fourth replication triggered abortion by accumulating 100 waiting jobs in the queue for machine 2 (results after 10 periods are shown in Table 68).

Close observation of the progress of any sequence of jobs after the initial warm up period showed "lumpy" or spasmodic use of route 2. For some time, route 2 was hardly used. Queues began to form on route 1 at machine 2. Finally the flowtime through route 1 increased enough for route 2 to report a lower flowtime. The longer sequence of operations delayed flowtime reporting, and while machine 2 cleared the route 1 backlog, further jobs were directed to route 2 until jobs started queuing through route 2 and the cycle reversed.

There was clearly an effect of the different operation times through routes 1 and 2 for a batch of 50 components. The feedback rule was amended to take account of

Table 67 Series 9: Partial routes, feedback method

Replication A

1. Machine use results	Machine no.			
	2	3	4	5
Mean % working	68.3	3.8	11.8	14.1
Mean % setting	12.0	0.3	0.3	0.3

2. Output	Component no.			
	1	2	3	4
Overall	1636	1068	880	634
Route 1	1584	1054	869	2
% batches thro' route 1	96.8	98.7	98.8	0.3

3. Flowtime	Component no.			
	1	2	3	4
Overall	982	1096	1163	786
Overall st.dev.	736	722	740	418
Maximum observed	3780	3687	3952	2161
Route 1	979	1092	1160	1116
Route 1 st.dev.	735	722	716	2118
Route 2	1091	1382	1379	785
Route 2 st.dev.	782	689	1894	410

Table 6A Series 9: Partial routes, feedback method

Replication D

1. Machine use results	Machine no. 2	3	4	5
Mean % working	65.7	11.0	30.6	21.3
Mean % setting	11.2	3.1	3.1	3.1

2. Output	Component no. 1	2	3	4
Overall	323	204	153	132
Route 1	241	155	92	108
% batches thro' route 1	75	76	60	82

3. Flowtime	Component no. 1	2	3	4
Overall	2111	2724	2730	2447
Overall st. dev.	1586	1863	1987	1665
Maximum observed	6834	7560	7717	7369
Route 1	2052	2453	2394	2231
Route 1 st. dev.	1587	1603	1670	1348
Route 2	2285	3579	3237	3422
Route 2 st. dev.	1638	2333	2310	2473

this difference at the time of comparison. Table 69 shows the results for the fourth replication which completed the run time without abortion. The flowtimes are still high and highly variable.

The following points are concluded from this section:

(a) Flowtime is very sensitive and cannot be easily used as a feedback variable

(b) Investigation is required into feedback and control theory to find out how feedback control may be effected

(c) The average flowtime indicators were not periodically zeroed or restarted in any form and successive completed jobs made less and less impact on the overall average.

In general, although the feedback results are poor here, this does not necessarily indicate inherent weakness in the method. It is considered that the method was not applied properly and deserves further investigation.

Table 69 Series 9: Partial routes, feedback method

Replication D - amended rule

1. Machine use results

	Machine no. 2	3	4	5
Mean % working	71.2	12.4	27.6	41.3
Mean % setting	10.5	0.9	0.9	0.9

2. Output

	Component no. 1	2	3	4
Overall	1663	1099	853	645
Route 1	1556	122	782	624
% batches thro' route 1	94	11	92	97

3. Flowtime

	Component no. 1	2	3	4
Overall	2004	1452	2426	2254
Overall st.dev.	1767	1472	1906	1812
Maximum observed	7330	7863	7705	6962
Route 1	1925	3233	2255	2227
Route 1 st.dev.	1682	2308	1734	1754
Route 2	3599	1230	4307	3035
Route 2 st.dev.	2537	1154	2595	3043

9.7 Series 10 - Partial routes, outer machines are the bottlenecks.

The data set used for series 6, 7 and 8 was altered to increase the operation times on the outer machines 1 and 6 by an arbitrary 65% for all components in order to make these machines the rate limiters. The operational data used for series 10 is presented in Table 70.

Ten replications of each of the decision rules in series 6 were made under exactly similar conditions to those used for series 6.

Table 71 shows the machine usage results for machine 1, 2, 4 and 5 in order to see how well each route was being used and to compare the level of loading before and after the alternate routing decision. It is observed immediately that under rule 2 the inner machine 2 continues to be more heavily loaded than the outer machine 1, despite the attempt to make the outer machine the bottleneck. Comparing Table 71 for series 10 with Table 55 for series 6 shows the heavier use of machine 2 in series 10 under all the adaptive rules.

The sequence of jobs leaving machine 1 in series 10 is identical using FCFS to the sequence in series 6. It is proposed that a similar effect to the effect observed in section 8.4 is operating here too. The relative lateness of completion on machine 1 compared with series 6 will result in a greater chance of zero WIP at machine 2 and hence more work will be routed to machine 2 (route 1). It is tentatively proposed that higher utilisation of the feeder machine will result in higher utilisation of route 1 under these rules, and WIP conditions.

Table 70 Operation data used for series 10

Comp	Op. No.	Op. time and capable machines 1	2	3	4	5	6
1	1	1.04					
	2		1.0				
	ALT2			0.7	+ 3.0	+ 1.5	
	3						1.04
2	1	1.55					
	2		2.0				
	ALT2			1.1	+ 2.3	+ 4.0	
	3						1.55
3	1	1.86					
	2		3.6				
	ALT2			1.6	+ 3.6	+ 1.0	
	3						1.86
4	1	2.57					
	2		4.0				
	ALT2			0.5	+ 1.5	+ 2.0	
	3						2.57

Other conditions:

Set-up times:	30 minutes
Inter-operational transport time:	30 minutes
Batch size:	50
Mean time to repair:	100 minutes
Mean time between failures	13.2 hours

Notes:

Machines 1 and 6 are most heavily loaded

Expected maximum machine utilisation is 65%

Table 71 Series 10: Partial routes, outer machines are bottleneck - Machine use results

	Machine no 1	2	4	5
1. Ratio rule				
Mean % working	65.5	51.6	43.2	33.6
Mean % setting	12.7	9.8	8.6	8.6
2. Least number of jobs in partial route				
Mean % working	65.5	67.1	38.6	29.5
Mean % setting	12.7	11.8	6.0	6.0
3. Least number of jobs before bottleneck				
Mean % working	65.5	63.2	42.8	32.6
Mean % setting	12.7	11.4	6.7	6.7
4. Least workload before bottleneck				
Mean % working	65.5	58.4	45.3	36.3
Mean % setting	12.7	10.4	6.9	6.9
5. Earliest expected start at bottleneck				
Mean % working	65.5	60.5	41.2	32.8
Mean % setting	12.7	10.9	7.0	7.0

In general, the percentages of work through route 1 in series 10 in table 72 are observed to be higher than those in Table 56 for series 6, in line with the higher machine utilisation.

Table 73 shows the flowtime results. The earliest expected start rule achieves the lowest flowtime for all components.

Table 72 Series 10: Partial routes, outer machines are bottleneck - Output results

	Component no. 1	2	3	4
Overall	1645	1085	898	646

1. Ratio rule

	1	2	3	4
Route 1	1234	727	449	213
Route 2	411	358	449	433
% batches thro' route 1	75	67	50	33

2. Least number of jobs in partial route

	1	2	3	4
Route 1	1031	723	636	486
Route 2	613	362	261	160
% batches thro' route 1	63	67	71	75

3. Least number of jobs before bottleneck

	1	2	3	4
Route 1	965	699	604	444
Route 2	678	386	293	201
% batches thro' route 1	59	64	67	69

4. Least workload before bottleneck

	1	2	3	4
Route 1	1047	609	540	406
Route 2	597	476	357	239
% batches thro' route 1	64	56	60	63

5. Earliest expected start at bottleneck

	1	2	3	4
Route 1	1146	657	541	410
Route 2	498	428	356	235
% batches thro' route 1	70	61	60	63

Table 73 Series 10: Partial routes, outer machines are bottlenecks

Flowtime results

	Component no. 1	2	3	4

1. Ratio rule

	1	2	3	4
Mean flowtime	1410	1612	1864	1949
St. Dev. of observ'ns	859	890	964	975

2. Least number of jobs in partial route

	1	2	3	4
Mean flowtime	1435	1608	1794	1821
St. Dev. of observ'ns	847	889	951	943

3. Least number of jobs before bottleneck

	1	2	3	4
Mean flowtime	1435	1608	1803	1834
St. Dev. of observ'ns	840	880	947	940

4. Least workload before bottleneck

	1	2	3	4
Mean flowtime	1418	1646	1834	1844
St. Dev. of observ'ns	851	901	964	955

5. Earliest expected start at bottleneck

	1	2	3	4
Mean flowtime	1388	1592	1799	1827
St. Dev. of observ'ns	849	898	973	960

9.8 Summary

When alternative routes include partial routes, rather than simply comprising single operation alternatives, the behaviour of simple decision rules is less predictable.

A "round robin" ratio rule which records previous allocation of jobs does not necessarily balance workload, if a different ratio is calculated for each component according to the processing capability of the rate limiting machine on each partial route. However, an overall ratio which is calculated using the workload expected on each machine will balance workload.

If flowtime is important, workload balance is not desirable. Lower overall flowtimes were achieved using the individual component processing capability ratios.

The earliest expected start rule performs consistently well and its success depends primarily on knowing the expected completion time of a job. Under this breakdown pattern, knowledge of any current repair time is much less important.

A simple feedback rule was tried and produced uneven flow and large flowtimes. However, it was considered that the method was not properly applied and there are no grounds for ignoring it in future work.

10. Final Discussion

The quality of the model, rules and results are now examined for their realism, practicality and application potential.

10.1 The Simulation Model

10.10.1 Job Arrival

The main control logic had been developed for a real company (79) where it is in regular use, although much of the Fortran code has been altered for the purposes of this work. One of the most important decisions when applying a model like this to research is the modelling of "arrivals" of jobs or orders. If schedules are being created, the vital questions are "How much data will be known before launch?" and "How many jobs may be examined before launch?", where launch onto the shop floor may be later than "arrival". Alternative methods for handling these questions which have been used in previous research were reported in section 2.18.1. Since schedules are not being constructed for groups of arriving jobs in this work, arrivals are simply launched into the shop immediately. However, a more fundamental but less discussed simulation issue is whether arrivals should occur at some average rate which is less than the processing rate, or whether jobs should be drawn into the model as required. All the methods in 2.18.1 can be used under either of these two processes. Unlimited arrivals are used to determine the maximum capacity or output rate of a system. An infinite buffer should not be allowed at the first processor unless that machine is the rate limiter. The size of the buffer allowed there and at other points in the system can have a significant impact on the output ability of the system, and investigation of this influence on the capacity should form part of the research task.

A limited arrival rate is usually used to test a set of conditions, as in this case, and infinite buffer storage is useful to understand where and why inventory accumulates.

10.1.2 Level of work-in-progress.

Although maximum machine working time was usually of the order of 65%, a realistic level, and minimum idle time was of the order of 15%, the average work in progress level was very low throughout the model and throughout the series. Maximum levels of 15-20 batches in progress were observed for each of components 1 and 2, although average levels for component 1 were 2-4 batches over the whole experimental period.

Because setting up and breakdown time are recorded separately from "idle" time , a machine is only idle for lack of material. In practice, machines may be idle due to lack of an operator or setter, during which time material can still accumulate, or material flow may be very "lumpy" or "bursty" causing long queues for short periods. The natural uneveness of material flow is caused in several ways in the model, for example, breakdowns, set up changes, forced process batch sizes and non-identical processing times.

The purpose of the work has been to test the effect of different decision rules on the flow of material through alternative routes. High work-in-progress, needing to be managed by despatching rules, would not have clarified this process. Clearly, these results are only valid for this utilisation level and development of more general results will require investigation at different utilisation levels, and consequent examination of how the alternate routing rules interact with despatching rules as they are introduced to manage work-in-progress.

10.1.3 Data

The operation times, set up times and inter-operational transport times are not untypical, although their uniformity is unrealistic. Further investigation is required to establish any dependency of these results on for example, the relative magnitude of inter-operational transport times to processing times, or processing times to setting up times. Similarly, the mean times between failure and mean repair times are not untypical, but different repair time patterns may yield different emphases. Here is the real weakness of the non-mathematical approach. It is difficult to prove the general application and benefits of the best rules from simulation of one set of conditions. On the other hand, practical experience emphasises the observations made in Chapter 2 that no one methodology is ever best under all circumstances and that some compromise is usually needed in practice to meet multiple objectives adequately in any one application. Hence complete generality may be an inappropriate goal.

Sufficient evidence has been gained from this work to conclude some of the different benefits of different rules. The strength of simulation has been to investigate the rules themselves, and test a wide variety of rules without compromising the repeatability of the model. The relative performance of the rules has been measured under similar conditions, and observations have been made on how the rules are working and how their performance is achieved.

10.2 Decision rules

Not all the rules proposed in section 5 were tested. A rule which chose the route which has the greatest current WIP of the component type in question was not tested on either single operation or partial route alternatives. Initial runs showed that the work-in-progress level was low enough to place great reliance on the secondary rule for breaking ties. Historical records could have been kept to push work on the route with most experience of this component type. To prevent ties being broken by the "first found" route without any experience of a component shortly after start up, a third rule, "round robin", was introduced to break start up ties. At this point, the nature of the overall decision was becoming considerably more complex and sophisticated than the other rules and its development was discontinued.

The work which has used this rule in practice (79) prefilled the model with jobs representing a typical level of work-in-progress and other typical start-up conditions were also entered, removing the need for the third rule. After finding this potential complexity for single operation alternatives, the rule was not tried on partial route alternatives. This rule should still be tested and could be included in a further comparison of hybrids, developed from some of the better rules already tested.

10.2.2 Cut-off point rule

The cut-off point rule, which has also been used in practice (63) was not tested here at all. Considerable experimentation is required to determine the best level for the cut-off point, to meet the conflicting objectives of sending as many jobs through the preferred processor as possible, while maintaining simultaneous workflow through as many processors as possible to meet demand. There must be some relationship between the cut-off point, average processing time and average utilisation level which could be developed in future work.

10.2.3 Earliest expected completion through alternate

As an outcome of the experimental work, a rule which is intuitively attractive but which would need great care in formulation, is to choose the partial route which offers the earliest expected completion time to the node at the end of the partial route branches. This rule is a natural development of the earliest expected start rule on single operation alternatives. Estimating the earliest expected finish time through single operation alternatives is straightforward, and it would be a valid criticism that "earliest expected finish" time would be more accurate than "earliest expected start" since a job may have significantly different processing rates on different processors. This criticism is also valid for both single operation alternatives and completion through the bottleneck machine in partial route alternatives.

However, on partial routes, the rate of arrival at the end of the partial route will depend on all the work-in-progress which has yet to go through the bottleneck plus any work in progress to go through later machines for which the decision job must subsequently wait. This total calculation is not impossible and indeed is the basis of forward scheduling, but can be very complex. In the interests of simplicity and visible effects again, the rule was not developed, and only the simple version "total number of jobs in partial route" was tested. Similarly, the "earliest expected start" rule allowed some comparison between effects on single operation alternates and partial route alternates.

10.2.4 Inclusion of jobs in transit

There was a lack of consistency in the counting of workload and jobs between the single operation alternatives series and the partial routes series between all rules used in both series, with the exception of the ratio rule. Jobs for which routing decisions have recently been made and which may be still in transit to the next processor were not included in the job or workload counts in the single operation alternatives series until they had arrived. This was changed in the partial route decision rules to include all jobs as soon as a decision on their destination had been made.

10.2.5 Feedback rule

The feedback rule was tested even though it had not been included in the formulation section, and the scant results were included even though they were far from complete.

These results were included to demonstrate the complexity of adapting well-tried principles to a rather sensitive measure and also to demonstrate the use of visual interactive simulation in understanding the flow of material which resulted from each rule.

The results are also interesting because they prove the need for two machines to carry out these operations (easily demonstrated by calculation) but the high flowtime results also demonstrate the efficacy of all the other rules in maintaining material flow and reducing flowtime. The wisdom of designating preferred routes through which all work should go unless there is a serious problem may be questioned by considering the efficiency of the rules tested, in comparison with the feedback results.

10.3 Dimensions of alternate routing which were not explicitly tested

10.3.1 Many to many nodes

In the introduction, it was stated that the node marking the starting point of the alternative routes could be of the type one to many or many to many. Even though only the one to many node has been examined, it is considered that the results are equally applicable to a many to many node so long as jobs for which decisions have just been made are included in all subsequent assessments. This inclusion is critical since jobs in transit from a number of sources could represent a significant volume of uncounted WIP and could bias concurrent decisions.

10.3.2 More than two options

Only two alternatives were tested in each series. No problems are envisaged in extending any of the rules tested to more than two options.

10.3.3 Unisymmetric or bisymmetric alternates

In the introduction, bisymmetric alternates were introduced where if B is an alternative to A, then A must be an alternative to B. This commutation is not necessarily true for unisymmetric alternates. This distinction should not have any importance for these rules where a decision is made for each transfer batch. The component and its next operation are identified, any alternates to that operation are identified and then a decision is made between the alternates available.

10.3.4 Preferred routes

Preferred routes were not explicitly tested but a number of observations may be made from the experimental results.

The ratio rule could clearly be used to push more work through the preferred route though it is likely that there will be a flowtime penalty. The proportions to be used by the ratio rule would need careful experimentation or calculation based on the capacity of the preferred route, taking into account breakdowns, setting up, tool changing and other diversions on the preferred route, in order to avoid overload.

Alternatively, any of the adaptive rules may be used as now but where ties should be broken onto the preferred machine. This will yield a relatively higher but unpredictable load on the preferred route.

Finally the adaptive rules could be implemented with some kind of workload counter, which would be a workload cut-off point rule on the preferred route.

10.3.5. Mixed capability machines and multiple machines

Machines of mixed capability will appear as alternatives at the decision point. Calculating appropriate ratios and counting appropriate workload for the adaptive rules are more complicated for machines of mixed capabilities, and are addressed in section 10.4.2.

10.4. Feasibility of operating these rules in practice

10.4.1 Data items required

The list of data required for the decision job and all rules is summarised in Table 74. Data required for the decision job is the same in all cases.

Knowledge of the exact progress of every job in each alternative route, in addition to the decision job, is required to obtain an accurate assessment of the number of jobs and their associated processing times for all of the adaptive rules. If a shop floor data capture system is being operated where operators log on to each new job, this information is readily available. Manual attempts at maintaining this information are tedious and inevitably inaccurate.

10.4.2. Calculation

The algorithms are all simple and speedy responses should be obtained to a destination enquiry made by either the operator who has just completed the work or a material mover who is available to move the recently completed batch.

Table 74 Data items required to operate these alternate routing rules

For the decision job	• Component type • Next operation number • Alternative operations available for next op no.
For all jobs	• Recorder of recent decisions
1. Ratio rule:	• Proportions to be sent to each alternative (i.e. the ratio) based on master schedule, operation times & batch sizes
2. Number of jobs in the route	• Operations list for all decision jobs on the route • List of WIP at these stages • Number of other machines capable of these operations
3. No. jobs before bottleneck	• Location of the bottleneck (by examining workload planned through each stage of the route) • Work-in progress at each alternative
4. Workload before bottleneck	• Processing time thro' bottleneck machine for each job in (3)
5. Earliest expected start at bottleneck	• As (4) plus remaining processing time on any current job being processed on bottleneck machine, plus any repair time estimate

Dispersed buffers have been assumed in the previous paragraph. If all the alternative machines are close together, the "first free machine" policy is simple to operate and is effective.

It is more complicated to assess the relative workload up to the bottleneck if one of the operations can be performed on more than one machine, either because there are multiple machines or machines of mixed capabilities. Some estimate can be made of the share of jobs which could be apportioned to one machine either by dividing the total workload for all the machines in a group or work centre by the number of machines, or by apportioning some share of the workload for each operation of which a machine is capable to that machine. Ultimately, the workload on the least loaded machine in the group may be compared with representative workloads on other machines in other groups on the partial route to find the bottleneck and also to estimate the workload up to the bottleneck. With an adequate shop floor data capture system, correct lists of capabilities of different machines and the operation times for different jobs on different machines, this calculation will be possible, even if it is not straightforward. Further work is required to establish the most effective method of calculation.

10.4.3 Position of bottleneck

During the experimental series, different data sets created a bottleneck inside the starting and finishing nodes of the alternative routes, or outside those nodes. One of the main purposes of using alternative routes is to eliminate a bottleneck (and inevitably move it elsewhere by doing so). Therefore the chance of the alternative routes remaining as the bottleneck is very small. These configurations were tested for experimental purposes only.

10.4.4 Use of dynamic routing

These alternate routing rules have been examined by making a routing decision based on the state of the network or by following some predetermined ratio. It is unlikely that jobs could be launched into a shop to find their own routes to completion, like data packets into a communications network. There is usually a schedule of expected completion times, and work-to lists issued daily or every 2-3 days finalise the schedule for each group of machines. It was stated in section 4.2 that some production management systems are considered to have progressed too far toward central control. Discussion with a production manager recently illustrated this. The scheduling system of an MRPII system scheduled the five operations for each job into five separate weeks. Each operation required between 30 and 120 minutes processing time for a batch. Lead times have now been cut dramatically by scheduling a job into a week, rather than an operation. It is not intended here to discuss bucketed and bucketless MRP or indeed to discuss the merits of MRPII systems but it is suggested that over complex control systems and over defined scheduling may be features of contemporary manufacturing control that should be examined. The role of expected lead times in MRPII systems is very much discussed (e.g. 64, 66) and perhaps local dynamic routing will play a major role in systems which operate input-output control correctly and which have good shop floor control systems. Dynamic routing in manufacturing depends on good data recording systems rather than operations planning systems.

10.5 Dynamic scheduling and dynamic routing.

Dynamic routing determines the next stage of the route at the time that a decision is required. In manufacturing, dynamic routing fixes the machine operation linkage by examining the different machines available for that operation at that time. Full dynamic routing considers only the next stage of manufacture at each decision.

Scheduling aims to create a machine-operation linkage for each known operation required of each known job, fixed to some time point within the sensible foreseeable future and subject to the constraints imposed by the sequential dependency of operations, availability of equipment, tools, operators and other resources. It was suggested that practitioners and researchers alike seek the one best schedule, but practitioners know that the schedule is unworkable in its entirety before it is issued because of the dynamism of the manufacturing environment. It may be that the focus of scheduling effort should change from producing the ultimate schedule to fast and smart rescheduling as suggested by Svestka (51).

Rather than produce a work-to list for a shift, a pool of potential jobs is proposed for processing, and a pool of potential jobs to be prepared is proposed. As each operator draws near the end of a job, an inquiry is made of the system for the most apporopriate job to be loaded next. By a short interactive session, the supervisor establishes with the system that the pack of jigs, fixtures, tools, drawings, etc. is available for the next jobs, and can investigate jobs expected to arrive shortly.

With high work-in-progress, visibility of material flow is poor, and complexity of ordering the manufacturing pack and components for machining increases dramatically. It is proposed that material flow should improve dramatically with alternate routing, that good alternate routing decisions depend on good shop floor work recording data, and shop floor data capture is vital for work-in-progress control.

10.6 Contribution of alternate routing to flexibility

"Increased flexibility" is a major contemporary manufacturing objective, spanning skills, budgeting and shop floor activities. "Economies of scale" are to be to replaced by "economies of scope" (88).

Due date flexibility for individual operations was mentioned in Chapter 4 and has been effected by producing "work-to" lists, equivalent to a local, short-term schedule, produced with the assistance of a local short-term capacity plan. Alternate routes may be exploited by either planning work onto different work centres and writing the alternate route into the routing sheet before job launch (effectively fixing the route), or by fixing the route into the work-to-list, or by determining each alternate decision (even if not the rest of the route) at the time the decision is required. So long as capacity planning can be achieved for these work centres, there is no reason why the routing decision cannot be made at the last minute to achieve maximum flexibility and to take advantage of information regarding the exact state of the network.

Operational flexibility, where operations on different components of the same type may be performed in different orders according to the state of the manufacturing network so long as any precedence relationships are maintained, is more difficult to achieve.

Full operational flexibility is easiest to achieve under fully dynamic routing. Although it might be envisaged that a group of jobs might converge on a machine which is finally repaired after a long breakdown, in general it must be expected that flowtimes would be low. The complexity of managing the movement of batches would be very much higher, approaching the job shop end of production layouts, when most objectives lie towards achieving a flow shop environment.

The advantages of FMSs, highly capable machines linked by random access materials handling equipment are clear. Operational flexibility may be achieved.

10.7 Assessment of the heuristics against objectives for good scheduling

A number of objectives for good scheduling were outlined in the introduction to chapter 5:

Minimise overall flowtime and prevent tails

Minimise overall machine utilisation

Minimise number of set up changes and hence total setting time

Minimise effect of delays and disturbances

Minimise queuing time

Minimise inventory

Minimise lateness

Minimise machine idle time

Due dates were not included in the model but some factors in due date achievement were examined. The industrial scheduler is looking for reproducability of flowtime such that there is confidence in achieving the due date at the time of launch. Spread in due date achievement increases with level of work in progress and the number of disturbances to flow, such as breakdowns and set up changes. Therefore the standard deviation of recorded flowtimes was included in the results as a measure of spread and commented upon where appropriate.

The maximum flowtimes did not vary greatly between rules and conditions. From the distribution of average, minimum and maximum flowtimes, the output distributions were clearly skewed, following the output distribution pattern recorded in figure 31. However, it is clear from the poor results of the feedback rule that alternate routing rules can be significant in flowtime management but that further work is required to establish the most important factors determining the spread of the output distribution and hence what particular effects there may be on due date achievement.

Fixing the routing to minimise the number of set up changes did reduce the total setting up time but was not particularly effective at reducing average flowtime. The other rules varied in their demand on set up time and tables 52 and 53 summarised the relatively poor performance of the ratio rule on non-identical machines but little difference was observed between the other rules. The ratio rule also incurred the highest total setting times in the partial route series.

Queuing time and inventory levels are closely linked to flowtime and hence efforts to reduce total flowtime will address these performance measures also.

To minimise machine idle time and minimise machine utilisation appear to be contradictory objectives. The aim was to use machines in the most effective way, by routing jobs to the fastest processor. If labour is very flexible and machines can be crewed for parts of a shift as required, it may be wise to route jobs to the fastest processor where possible. However, it appears from this work that flowtime penalties may be incurred if alternate routes are not managed well. Where labour is not flexible and machines will be crewed even if work is not always available, adherence to a good alternate routing rule, e.g. earliest expected start, will assist attainment of lower flowtimes overall and hence schedule achievement.

The effect of using alternate routes to counter delays and disturbances was not measured. A proposal that routing flexibility in FMS should be measured by the "robustness" of the system to breakdowns, by assessing how well the production rate was maintained, was reported in section 3.4 (40). This type of measurement could be an item for further work. Total "flexibility" according to this measurement does indeed imply provision of spare capacity as Falkner (41) observed. Experience from other simulation projects leads the author to conclude that buffer stock levels and reassessment of short term product mix can have an important role in recovery, without needing excessive spare capacity.

10.8 Assessment of the heuristics against objectives for good heuristics

Four objectives for good heuristics were stated by Silver, Vidal and de Werra (37) and listed in section 2.19:

(a) Realistic computational effort to obtain the solution

(b) The solution should be very close to the optimum on average

(c) The chance of a very poor solution should be low

(d) The heuristic should be understandable by the user, preferably explainable in intuitive terms

In addition to these qualities, "minimum information requirement" is added in chapter 5, where (a) above is taken to refer to the complexity of the algorithm, and the information requirement refers to the number of data items and files which must be accessed.

The data requirements were discussed in section 10.4.1. It is considerd that these requirements are not onerous for a shop floor work reporting system. The heuristics tested were not complex. Where secondary rules are required, the processing time may be increased but local decisions from locally constrained data should yield a fast decision.

With regard to the optimality of the decision, it has already been suggested that "optimal" is not necessarily an appropriate term for practical scheduling with regard to the multiplicity of objectives and the dynamism of the environment. Different rules should be employed for different objectives according to local configurations and demands. Different rules may be employed at different decision nodes in the manufacturing network. Poor decisions may be avoided by avoiding the poorer rules and by not employing rules which do not yield good results in certain situations.

It is believed that all of the rules tested are understandable and explainable in intuitive terms.

11. Conclusions

1. A generic simulation model was constructed which allowed different routing files to be tested on a simple machine shop. Breakdowns, setting up times and inter-operational transport times were included in the model.

2. Rules were formulated to examine the different ways in which material could be directed through alternate routes.

3. The alternate routing rules were tested on single operation alternates and partial route alternates. Performance was measured by batch flowtimes, the number of batches of output and the amount of working and setting up time on critical machines.

4. In the simple cases of machines operating within close range of one another, where they may draw work from the same buffer, the efficacy of the "first free machine" rule, proposed for FMSs, has been confirmed.

5. However, where machines are dispersed in a shop, and a routing decision is required, it has been shown that the rule used to make that decision can have a significant effect on machine use, total setting time and average flowtime. As a result, average inventory levels and queue lengths will be affected. Where partial routes are involved, the decision becomes even more significant.

6. The workload balance concept has not been found to be effective where flowtimes are an important consideration, although workload balance can be achieved for adaptive and non-adaptive rules if equable work distribution is a priority.

7. There is an apparent contradiction in the trend for increased flexibility, which would favour increased use of alternative routes, and the drive to simplify manufacturing routes by implementing cells dedicated to groups or families of parts.

12. Recommendations for further work

Further work arises from the experimental work and from the final discussion.

12.1 Further work arising from the experimental work

The following issues remain unresolved from this experimental work:

12.1.1 Effect of inter-operational transport time on the quality of the decision, especially with regard to the magnitude of the average processing time.

12.1.2 Effect of the breakdown level and repair time patterns on :

(i) the quality of the earliest expected start rule and the importance of including repair time in the workload assessment

(ii) the relative performance of the non-adaptive rules and the adaptive rules, where the adaptive rules should maintain their performance when machines have non-identical breakdown patterns and high breakdown levels.

12.1.3 Development of a feedback rule, probably based on flowtime.

12.1.4 Development of a ratio rule which records projected workload despatched to each route rather than the number of jobs, and aims to balance workload (50%:50%) rather than maintain a ratio derived from an expected wokload on each machine. This new rule will only need to identify which is the bottleneck machine in each route.

12.1.5 Effect of despatching rules at higher WIP levels on these results.

12.2 Further work arising from the final discussion

12.2.1 Dependence of these results on the utilisation level of the equipment.

12.2.2 Effect of process batch size on the quality of the decision, and the relative magnitude and variability of the setting up time on the decisions.

12.2.3 Development of the "greatest current WIP in the route" rule, and particularly how "experience" of an operation is recorded.

12.2.4 Investigation of the principal parameters in a workload cut-off point rule, and particularly how it may be generalised into a relationship including processing rate, number of alternatives available, and importance of following the preferred route.

12.2.5 Any difference arising from assessing earliest completion instead of earliest expected start, and how earliest expected completion through a partial route may be gauged in a straightforward manner.

12.2.6 Effect of alternate routing rules on due date flexibility

12.2.7 How "routing flexibility" may be measured

REFERENCES

1. Franks, Stephen, Scheduling Technology Group, Hounslow, Middx. (Private communication).

2. Sobczak, T.V. (ed.) " Glossary of Terms for Computer Integrated Manufacturing", Society of Manufacturing Engineers, 1984.

3. Russo, F.J. "A Heuristic Approach to Alternate Routing", Master's thesis, MIT, June 1965.

4. Wayson, R.D. "The Effect of Alternate Machines on Two Priority Dispatching Disciplines in the General Job Shop", Master's thesis, Cornell, February 1965.

5. Gere, W.S. "A Heuristic Approach to Job Shop Scheduling", unpublished PhD thesis, Carnegie Institute of Technology, September 1962.

6. Mellor, P., "A Review of Job Shop Scheduling", Operations Research Quarterly, Vol. 17 (2) 1966, p. 161-171.

7. Sisson, R.L., "Methods of Sequencing in Job Shops - A Review", Operations Research, Vol. 7(1), 1959, p.10-29.

8. Sisson, R.L., "Sequencing Theory", Chapter 7 in "Progress in Operations Research", ed. R.L. Ackoff, Wiley, 1961.

9. Blackstone, J.H., Phillips, D.T., Hogg, G.L., " State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operation", Int. J. Prod. Res. Vol. 20(1), 1982, p.27-45.

10. Beenhakker, H.L., "Development of Alternate Criteria for Optimality in the Machine Sequencing Problem", PhD thesis, Purdue University, 1963.

11. Conway, R.W., Maxwell, W.L., Miller, L.W., "Theory of Scheduling", Addison-Wesley, 1967.

12. Graves, S.C., "A Review of Production Scheduling", Operations Research, Vol. 29(4), 1981, p. 646-675.

13. Smith, W.E., "Various Optimizers for Single Stage Production", Naval Research and Logistics Quarterly, Vol. 3 (1), 1956, p. 59-66.

14. Jackson, J.R., "Scheduling a Production Line to Minimize Maximum Tardiness", Research Report No. 43, Management Sciences Research Project, University of California at Los Angeles, January 1955.

15. Panwalkar, S.S., Iskander, W., "A Survey of Scheduling Rules", Operations Research, Vol. 25(1), 1977, p.45-61.

16. Kiran, A.S., Smith, M.L., "Simulation Studies in Job Shop Scheduling - II : Performance of Priority Rules", Computers and Industrial Engineering, Vol. 8(2), 1984, p. 95-105.

17. Gere, W.S., "Heuristics in Job Shop Scheduling", Management Science, Vol. 13(3), 1966, p. 167-190.

18. Coffman, E.G. (Ed.), "Computer and Job Shop Scheduling Theory", Wiley, 1976.

19. French, S., "Sequencing and Scheduling", Ellis Horwood, 1982.

20. Johnson, S.M., "Optimal Two and Three Stage Production Schedules with Set Up Times Included", Naval Research and Logistics Quarterly, Vol. 1, 1954, p. 61-68, reprinted in "Industrial Scheduling", ed. J.F. Muth and G.L. Thompson, Prentice-Hall, 1963, p. 13-20.

21. Akers, S.B., "A Graphical Approach to Production Scheduling Problems", Operations Research, Vol. 4 (2), 1956, p. 244-245.

22. Bellman, R., "Dynamic Programming", Princeton University Press, 1957.

23. Taha, H.A., "Operations Research", Macmillan, 1982.

24. Baker, K.R., "A Comparative Survey of Flow Shop Algorithms", Operations Research, Vol. 23(1), 1975, p.62-73.

25. Jackson, J.R., "Networks of Waiting Lines", Operations Research, Vol. 5(4), 1957, p.518-521.

26. Suri, R., "An Overview of Evaluative Models for Flexible Manufacturing Systems", Annals of O.R. Vol. 3, 1985, p. 13-21.

27. Marchal, W.G., "Numerical Performance of Approximate Queuing Formulae With Application to Flexible Manufacturing Systems", Annals of O.R., Vol. 3, 1985, p. 141-152.

28. Seidmann, A., Shalev-Oren, S., Schweitzer, P.J., "An Analytical Review of Several Computerised Queuing Network Models of F.M.S", Proc. 2nd ORSA/TIMS Conf. on FMS. ed. K.E. Stecke and R. Suri, pub Elsevier, 1986.

29. Proc. 1st Int. Conf. on "Simulation in Manufacturing", SIM-1, Stratford, March 1985, ed. W.B. Heginbotham, pub. IFS.

30. Proc. 2nd. Int. Conf. on "Simulation in Manufacturing", SIM-2, Chicago, June 1986, ed. J.E. Lenz, pub. IFS.

31. Proc. 3rd Int. Conf. on "Simulation in Manufacturing", SIM-3, Turin, November 1987, ed. G.F.Micheletti, pub. IFS.

32. Proc. 4th Int. Conf. on "Simulation in Manufacturing", SIM-4, London, 1988, ed. J.Browne, K.Rathmill, pub. IFS.

33. Hurrion, R.D., "The Design, Use and Required Facilities of an Interactive Visual Computer Simulation Language to Explore Production Planning Problems", PhD thesis, University of London, 1976.

34. Hurrion, R.D., "An Interactive Visual Simulation System for Industrial Management", European Journal of O.R., Vol. 5(2), 1980, p. 86-93.

35. Kiran, A.S., Smith, M.L., "Simulation Studies in Job Shop Scheduling : A Survey", Proc. Conf on Simulation in Inventory and Production Control, 1983, San Diego, California, ed. H. Bekiroghu, pub. Society for Computer Simulation.

36. Ashour, S., Vaswani, S.D., "A GASP Simulation Study of Job Shop Scheduling", Simulation, Vol. 8, 1972, p.1-10.

37. Silver, E.A., Vidal, R.V.V. , de Werra, D., "A Tutorial on Heuristic Methods", European Journal of O.R., Vol. 5 (3), 1980, p. 153-162.

38. Nicholson, T., "Optimisation in Industry", Vol. 1, Optimisation Techniques, Longman, 1971 (Chap. 10).

39. Neimeier, H.A., "An Investigation of Alternate Routing in a Job Shop", Master's Thesis, 1967, Cornell.

40. Browne, J., Dubois, D., Rathmill, K., Sethi, S., Stecke, K., "Classification of Flexible Manufacturing Systems", Section 2.2 in "Manufacturing Systems: Context, Applications and Techniques", ed. Bignell et al, Open University, 1985, originally appeared in The FMS Magazine, April, 1984.

41. Falkner, C.H., "Flexibility in Manufacturing Plants", Proc. 2nd ORSA/TIMS Conf. on Flexible Manufacturing Systems,, pub. Elsevier, 1986.

42. Bhattacharyya, S.K., Roy, R., Huang Y. S., "Knowledge Based Simulation Techniques for Control of FMS", Proc. Beijing Int. Conf. on System Simulation and Scientific Computing, Vol.3, 1989.

43. Fox, M.S., "Constraint Directed Search : A Case Study of Job Shop Scheduling", Pitman, 1987.

44. Maimon, O.Z., Choong, Y.F., "Dynamic Routing in Reentrant FMS", Proc. 2nd ORSA/TIMS Conf. on Flexible Manufacturing Systems, pub. Elsevier, 1986.

45. Cassandras, C.G., "Dynamic Material Handling for a Class of Flexible Manufacturing Systems", Annals of O.R., Vol. 3, 1985, p. 427-447.

46. Yao, D.D., Buzacott, J.A., "Modeling a Class of State-Dependent Routing in Flexible Manufacturing Systems", Annals of O.R., Vol. 3, 1985, p. 163-167.

47. Hahne, E.L., "Dynamic Routing in an Unreliable Manufacturing Network with Limited Storage", Laboratory for Information and Decision Systems, MIT, Report No. LIDS-TH-1063, February 1981.

48. Whitt, W., "The Queuing Network Analyzer", Bell System Technical Journal, Vol. 62(9), Nov. 1983, p. 2779-2815.

49. Segal, M., Whitt, W., "A Queuing Network Analyzer for Manufacturing", Proc. 12th Int. Teletraffic Congress, Torino, Italy, June 1988.

50. Chang, Y.L., Sullivan, R.S., Bagchi, U., "Experimental Investigation of Real-Time Scheduling in Flexible Manufacturing Systems", Annals of O.R., Vol. 3, 1985, p. 355-377.

51. Svestka, J.A., "A Real-Time Rescheduler-Supplying the Missing Link", Recent Developments in Production Research, ed. A. Mital, pub. Elsevier, 1988.

52. Stecke, K.E., Solberg, J.J., "Loading and Control Policies for a Flexible Manufacturing System", Int. J. of Production Research, Vol. 19(5), 1981, p.481-490.

53. Bell, P.R., Jabbour, K, "Review of Point-to-Point Network Routing Algorithms", IEEE Communications Magazine, Vol. 24 (1), January 1986, p.34-39.

54. Gallager, R.G., "A Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Transactions on Communications, Vol. COM-25(1), January 1977, p. 73-85.

55. Segall, A., "The Modeling of Adaptive Routing in Data Communications Networks, IEEE Transactions on Communications, Vol. COM-25(1), January 1977, p. 85-95.

56. Bell, P.R., Jabbour, K., "Evaluation of Point-to-Point Network Routing Algorithms", IEEE Transactions on Communications, Vol. COM-35(4), April 1987, p. 470-472.

57. Ephremides, A., Varaiya, P., Walrand, J., "A Simple Dynamic Routing Problem", IEEE Transactions on Automatic Control, Vol. AC-25(4), August 1980, p. 690-693.

58. Foschini, G.J., Salz, J., "A Basic Dynamic Routing Problem and Diffusion", IEEE Transactions on Communications, Vol. COM-26(3), March 1978, p. 320-327.

59. Chen, P., Talavage, J., "Production Decision Support Systems for Computerised Manufacturing Systems", Journal of Manufacturing Systems, Vol. 1(2), 1982, p. 157-167.

60. Nof, S.Y., Barash, M.M., Solberg, J.J., "Operational Control of Item Flow in Versatile Manufacturing Systems", International Journal of Production Research, Vol. 15 (5), 1979, p. 479-489.

61. Schlauch, R., "Production Scheduling in a Flexible Manufacturing System (FMS)", Proc. 1st Int. Conf. on Flexible Manufacturing Systems, London, 1982, pub. IFS.

62. Manara, R., Guiffre, O., Cavagnaro, F., "A Generalised Approach to the Problems of FMS On-Line Management", Proc. 1st. Int. Conf. on Flexible Manufacturing Systems, London, 1982, pub. IFS.

63. Grinsted, S.E., "Simulation of Mixing Areas at Avon Rubber Limited", University of Warwick Manufacturing Systems Engineering Group Report, March 1989 (unpublished).

64. Vollmann, T.E., Berry, W.L., Whybark, D.C., "Manufacturing Planning and Control Systems", Irwin, 1984.

65. Orlicky, J., "Material Requirements Planning", McGraw Hill, 1975.

66. Blackstone, J.H., "Capacity Management", APICS and South Western, 1989.

67. Sartori, L.G., "Manufacturing Information Systems", Addison-Wesley, 1988.

68. Kochhar, A.K. "Development of Computer Based Production Systems", Edward Arnold, 1979

69. Butcher, R., Production Manager, Lucas Engine Management Systems, Birmingham, Private Communication.

70. Groover, M.P., Zimmers Jr., E.W., "CAD/CAM : Computer Aided Design and Manufacturing", Prentice-Hall, 1984.

71. Davidson, G.J.F., "Computer Aided Process Planning and Standard Time Estimating at Lucas Aerospace Power Systems Division", Master's Dissertation, University of Warwick, September 1989.

72. Lundigran, R., "What Is This Thing Called OPT?", Production and Inventory Management, Second Quarter, 1986, p. 2-11.

73. Anon., "Overview of the OPT21 Software", Scheduling Technology Group Limited, Hounslow.

74. Harrison, M.C., "The Concepts of Optimised Production Technology (OPT)", Proc. 20th European Tech. Conf. on Production and Inventory Control, Birmingham, 1985, p.15-28.

75. Black, J., "Capacity Planning System", Section 3.2 in "Manufacturing Systems: Context, Applications and Techniques", ed. Bignell et al. Open University, 1985, originally appeared as "Developing Your Own Capacity Planning System" in The Production Engineer, March 1984.

76. Bakewell, D., Production Control Manager, Westland Helicopters, Private Communication, March 1990.

77. Carter, M.F., "Designing Flexibility Into Automated Manufacturing Systems, Proc. 2nd ORSA/TIMS Conf. on FMS, ed. K.E. Stecke and R. Suri, pub. Elsevier, 1986.

78. Bhattacharyya, S.K., Roy, R., McCarthy, P.A., Meikle, S., "The Use of Computer Simulation in Operational Planning and Control of Press Shops", ACME Research Conf., Birmingham University, September 1990.

79. Roy, R., "Implementation of a Factory Planning Model at Bajaj Auto Limited, Pune, India", Department of Engineering, University of Warwick, August 1987, (unpublished).

80. "See - Why User Reference Manual", Istel, Redditch, Worcs.

81. Carson, J.S., "Simulation Series, Part 2: Convincing Users of Model's Validity is Challenging Aspect of Modeler's Job", Industrial Engineering, Vol. 18(6), June 1986, p. 74-85.

82. Law, A.M., Kelton, W.D. "Simulation Modeling and Analysis", McGraw-Hill, 1982.

83. Banks, J., Carson, J.S., "Discrete-Event Systems Simulation", Prentice-Hall, 1984.

84. Pidd, M., "Computer Simulation in Management Science", Wiley, 1984.

85. Welch, P.D., "The Statistical Analysis of Simulation Results" in "The Computer Performance Modeling Handbook", ed. S.S. Lavenberg, p. 268-328, Academic Press, 1983.

86. Saliby, E., "A Reappraisal of Some Simulation Fundamentals", PhD thesis, University of Lancaster.

87. Fry, T.D., Smith, A.E., "A Procedure for Implementing Input/Output Control: A Case study", Production and Inventory Management, Fourth Quarter, 1987

88. Perez, C., "Micro-electronics, Long Waves and World Structural Change", World Development, Vol 13(3), 1985

Bibliography

1. Bellman, R., Esogbue, A.O., Nabeshima, I., "Mathematical Aspects of Scheduling and Applications", 1982, Pergamon

2. Jaffe, J.M., Baratz, A.E., Segall, A., "Subtle Design Issues in the Implementation of Distributed, Dynamic Routing Algorithms", Computer Networks and ISDN Systems, 1986, Vol 12(3), p147-158

3. Ash, G.R., Crafton, R.G., Timm, P.E., "Building a Dynamic Routing Architecture", Telephony, 1983, Vol 204(26), p42-53

4. Frank, H., Chou, W., "Routing in Computer Networks", Networks, 1971, Vol 1(2), p99-112

5. Goldratt, E.M., Cox, J., "The Goal", Creative Output Books, 1986

6. Swann, D.,"Using MRP for Optimised Schedules (Emulating OPT)", Production and Inventory Management, Second Quarter, 1986, p30-37

7. Meleton Jr, M.P., "OPT - Fantasy or Breakthrough?", Production and Inventory Management, Second Quarter, 1986, p13-21

8. Vollmann, T.E., "OPT as an Enhancement to MRPII", Production and Inventory Management, Second Quarter, 1986, p38-47

9. Plenert, G., Best, T.D., "MRP, JIT and OPT: What's Best?", Production and Inventory Management, Second Quarter, 1986, p22-29

10. Law, A.M., 'Statistical Analysis of Simulation Output Data ", Operations Research, Vol 31(6), p983-1029, Nov-Dec 1983

11. Kelton, W.D., "Simulation Series, Part 5: Statistical Analysis Methods Enhance Usefulness, Reliability of Simulation Models", Industrial Engineering, Vol 18(9), p74-86, September 1986

12. Hayslett, H.T., "Statistics Made Simple", Heinemann, 1988

13. Kheir, N.A., Damborg, M., Zucker, P., Lalwani, C.S., "Credibility of Models (Panel Discussion)", Simulation, August 1985, p87-89

Appendix 1

1.1 See Why Executive and Main Program

```
C______________________________________________________________
C
C   ALTERNATIVE ROUTING - SUE GRINSTED
C______________________________________________________________

C See Why executive
      SUBROUTINE OWNCY
      INCLUDE SCM91.FOR

      CALL DEFINE
      CALL TIMEON(2)
C Prefill :wip
      CALL PREFIL

C Initialise
      CALL INITAL
      CALL INITL2
      CALL ILDON
      CALL REFORM(0)

C Time advance
15    CALL ONWARD(NEXTE,NEXTM)
      GOTO(10,20,30,40,50,60,70,80,90,100,110,120),
     & NEXTE

C Arrival of pallets
10    CALL ARRIVE(NEXTM)
      GOTO 15

C End movement between groups
20    CALL ENDMOV(NEXTM)
      CALL STARCM(NEXTM)
      GOTO 15

C Set up complete
30    CALL SETUPC(NEXTM)
      CALL MCNUMB(NEXTM,MACHNO,IC,IG,IM)
      CALL STARMC(MACHNO)
      CALL SETUPS(0)
      GOTO 15

C Start breakdown
40    CALL BRDOWN(NEXTM)
      GOTO 15

C End breakdown
50    CALL UPMACH(NEXTM)
      CALL MCNUMB(NEXTM,MACHNO,IC,IG,IM)
      IH=IHEDOF(IMACH(MACHNO))
      CALL STPROC(IH,MACHNO)
      GOTO 15

C End processing
60    CALL ENDPRC(NEXTM,MACHNO)
      CALL CONEND(NEXTM,MACHNO)
      GOTO 15

C Start tool change
```

```
70     CALL TLCHGS(NEXTM)
       GOTO 15

C End tool change
80     CALL TLCHGF(NEXTM,IH,MACHNO)
       CALL STPROC(IH,MACHNO)
       GOTO 15


C End of day/shift
90     CALL ENDAY
       GOTO 15

C End of warm-up period
100    CALL ENDWRM
       GOTO 15

C End of experiment
110    CALL REPSUM
       CALL ENDEXP
       GOTO 15

C End of simulation
120    CALL ENDSIM
       GOTO 15

       END

C_______________________________________________________
C
C   INITIALISE COMMON VARIABLES
C_______________________________________________________
C
       BLOCK DATA INFO
       INCLUDE SCM91.FOR

       DATA STATE /'1', '2', '3', '4', '5', '6', '7',
     &'8', '9', '0'/

       END

C_______________________________________________________
C
C   OWN INTERACTIONS
C_______________________________________________________
C
       SUBROUTINE OWNINT(INT)
       INCLUDE SCM91.FOR

C Turn off current displays
       DO 10 I=1,9
         CALL DSPOFF(I)
         CALL DSPOFF((NOPGE*10)+I)
10     CONTINUE

C Turn on displays for interaction screens
       DO 20 I=8,10
         CALL DISPON(I,1)
```

```
20     CONTINUE

       CALL REFORM(0)
       GOTO (100,200,300,400,500,600),INT

C Change parameters : PRAM
100    CALL PARAM
       GOTO 99

C Change shop currently displayed : CDSP
200    CALL SHPDSP
       GOTO 99

C Display machine status/time results : MCHQ
300    CALL UPDATM
       CALL MCHQRY
       GOTO 99

C Display production so far : PRDQ
400    CALL PRDQRY
       GOTO 99

C Lead time breakdown by component type : LDTM
500    CALL UPDWIP
       CALL LDTIME
       GOTO 99

C Print reports on demand e.g. to test : REPT
600    CALL REPSUM
       GOTO 99

C Return to shop display : BACK
99     CALL BACKDS

999    RETURN
       END
C_____________________________________________________
C
C   SET UP DISPLAY OF MACHINES
C_____________________________________________________
C
       SUBROUTINE MCHDSP(MACH,ISTATE)
       INCLUDE SCM91.FOR

       CALL MCNUMB(MACH,MACHNO,I1,I2,I3)
       NOLD=IATTRB(MACH,2)

       IX=KXCORD(I1,I2)+I3-1
       IY=KYCORD(I1,I2)
       IS=IATTRB(MACH,1)
       IF(IS.LE.0) IS=41

C Change display
       CALL ISETAT(MACH,2,ISTATE)
       IZ=KPGENM(I1)

       IF (IZ .EQ.NOPGE) THEN
```

```
          CALL CVAREA ((NOPGE*10)+2+ISTATE, IX, IX, IY,
     &    IY, STATE(IS))
        ENDIF
C Update machine utilisation recorders
        WTIME=STIME(1)-RMCOP(MACHNO,3)
        RMCOP(MACHNO,3)=STIME(1)
        RMCOP(MACHNO,3+NOLD)=RMCOP(MACHNO,3+NOLD)+WTIME
        RETURN
        END

C_____________________________________________________________
C
C  RETURNS MACHINE NUMBER FROM CELL, GROUP & MACHINE
C  NUMBER
C_____________________________________________________________

      FUNCTION MCFIND(IC,IG,IM)
      INCLUDE SCM91.FOR

      NUM=0
      DO 50 I=1,IC-1
        DO 40 J=1,KNOGRP(I)
          NUM=NUM+KNOMCH(I,J)
40      CONTINUE
50    CONTINUE

      DO 60 J=1,IG-1
        NUM=NUM+KNOMCH(IC,J)
60    CONTINUE
      MCFIND=NUM+IM

      RETURN
      END

C_____________________________________________________________
C
C RANDOM NUMBER GENERATOR
C_____________________________________________________________

      SUBROUTINE RNDNUM(INT,MACHNO,U)
      INCLUDE SCM91.FOR
C If int=0 then breakdown, if int=1 then find repair
C time class, if int=2 then repair time
      IS1=KSEED(MACHNO,1+(INT*3))
      IS2=KSEED(MACHNO,2+(INT*3))
      IS3=KSEED(MACHNO,3+(INT*3))
      IS1=171*MOD(IS1,177)-2*(IS1/177)
      IF(IS1.LT.0)IS1=IS1+30269
      IS2=172*MOD(IS2,176)-35*(IS2/176)
      IF(IS2.LT.0)IS2=IS2+30307
      IS3=170*MOD(IS3,178)-63*(IS3/178)
      IF(IS3.LT.0)IS3=IS3+30323
      U=AMOD((FLOAT(IS1)/30269.+FLOAT(IS2)/30307.+FLOAT
     &(IS3)/30323.),1.)
      KSEED(MACHNO,1+(INT*3))=IS1
      KSEED(MACHNO,2+(INT*3))=IS2
      KSEED(MACHNO,3+(INT*3))=IS3
```

```
      RETURN
      END

C_____________________________________________________________
C
C    RETURNS MACHINE NUMBER
C    INPUT:  MACH=INTERNAL POINTER
C    OUTPUT: MACHNO=MACHINE NUMBER, IC=CELL, IG=GROUP,
C    IM=M/C#
C
C_____________________________________________________________
      SUBROUTINE MCNUMB(MACH,MACHNO,IC,IG,IM)
      INCLUDE SCM91.FOR

      MACHNO=0
      DO 20 I=1,NOCLS
        IC=I
        NO=KNOGRP(I)
        DO 10 J=1,NO
          IG=J
          NO2=KNOMCH(I,J)
          DO 5 K=1,NO2
            IM=K
            MACHNO=MACHNO+1
            IF(MACH.EQ.JMACH(MACHNO)) RETURN
5         CONTINUE
10      CONTINUE
20    CONTINUE

      END

C_____________________________________________________________
C
C Sort NO record pointers (INM) according to given
C integer values (IMUCH)
C_____________________________________________________________

      SUBROUTINE SORT(NO,INM,IMUCH)
      INCLUDE SCM91.FOR
      DIMENSION INM(NO),IMUCH(NO)

C Sort
         DO 20 I=1,NO-1
            DO 15 J=I+1,NO
               K=INM(I)
               L=INM(J)
               IF(IMUCH(K).GT.IMUCH(L)) THEN
                  INM(I)=L
                  INM(J)=K
               ENDIF
15          CONTINUE
20       CONTINUE
      RETURN
      END

C_____________________________________________________________
C
```

```
C   UPDATE COMPONENT STATE RECORDERS
C______________________________________________________________

      SUBROUTINE UPDATC(IH,ISTATE)
      INCLUDE SCM91.FOR

C Update component recorders
      NCOMP=IATTRB(IH,1)
      NOLD=IATTRB(IH,5)
      WTIME=STIME(1)-RATTRB(IH,6)
      RCMOP(NCOMP,NOLD+1)=RCMOP(NCOMP,NOLD+1)+WTIME

C Increment total recorder
      RCMOP(NCOMP,1)=RCMOP(NCOMP,1)+WTIME

C Reset entity attributes
      CALL ISETAT(IH,5,ISTATE)
      CALL RSETAT(IH,6,STIME(1))

      RETURN
      END

C______________________________________________________________
C
C   DISPLAY HOURS AND MINUTES ON SCREEN
C______________________________________________________________

      SUBROUTINE OWNTIM(TIME)
      INCLUDE SCM91.FOR

      NHRS=INT(TIME/60.)
      RMINS=TIME-FLOAT(NHRS)*60.
      CALL HINT(100,41,48,NHRS,6)
      CALL HTEXT(100,48,48,'HRS')
      CALL HREAL(100,53,48,RMINS,5,2)
      CALL HTEXT(100,59,48,'MINS')
      RETURN
      END

C______________________________________________________________
C
C ZERO OFF ATTRIBUTES OF PALLET ENTITY
C______________________________________________________________

      SUBROUTINE ZERPLT(IH)
      INCLUDE SCM91.FOR

      CALL ISETAT(IH,1,0)
      CALL ISETAT(IH,2,0)
      CALL RSETAT(IH,3,0.0)
      CALL RSETAT(IH,4,0.0)
      CALL ISETAT(IH,5,0)
      CALL RSETAT(IH,6,0.0)
      CALL ISETAT(IH,7,0)

      RETURN
      END
```

Appendix 1.2

Overlay 1 - Work Control

```
C______________________________________________________________
C______________________________________________________________
C MOVE FROM DUMMY TO NEXT MACHINE GROUP
C______________________________________________________________

      SUBROUTINE ENDMOV(JPLT)
      INCLUDE SCM91.FOR

C Find comp no. and next op no.
      NCOMP=IATTRB(JPLT,1)
      NXTOP=IATTRB(JPLT,2)

C From the opcode, find the next cell and group no.
      NCELL=KOPCEL(NCOMOP(NCOMP)+NXTOP)
      NGRP=KOPGRP(NCOMOP(NCOMP)+NXTOP)

C Move to next cell and group
      CALL ENDMV1(JPLT,NCELL,NGRP)
      CALL ADLAST(JPLT,IFLOOR(NCELL,NGRP))

C Update component state recorder
      ISTATE=1
      CALL UPDATC(JPLT,ISTATE)

      RETURN
      END

C______________________________________________________________
C
C   End move for current pallet and Schedule next in
C   dummy
C   Input:  IH = Pallet to be removed from dummy
C______________________________________________________________

      SUBROUTINE ENDMV1(IH,NC,NG)
      INCLUDE SCM91.FOR

      IDUM=IDUMMY(NC,NG)
      NDUM=ISIZOF(IDUM)

      IF(NDUM.LE.0) THEN
         CALL BTEXTI('ERROR IN SCHMOVE',NC*100+NG,5)
         CALL INTRCT
      ENDIF

      CALL RSETAT(IH,3,0.0)
      CALL REMOV(IH,IDUM)

      NDUM=ISIZOF(IDUM)
      IF(NDUM.GT.0) THEN
         T1=STIME(1)
         T2=99999.9
C Schedule end move on the entity due to finish moving
C next
         DO 100 N1=1,NDUM
            IH1=IDNTOF(IDUM,N1)
            T3=RATTRB(IH1,3)-T1
C Identify if an entity is due to arrive now
```

```
            IF(T3.LE.0.0) THEN
               CALL RSETAT(IH1,3,0.0)
               CALL SCHEDL(2,T3,IH1)
               GOTO 999
            ENDIF
            IF(T3.LT.T2) THEN
               T2=T3
               IHNXT=IH1
            ENDIF
100       CONTINUE
          CALL RSETAT(IHNXT,3,0.0)
          CALL SCHEDL(2,T2,IHNXT)
      ENDIF

999   RETURN
      END

C______________________________________________________________________
C
C Consequential event routine to MOVE and ENDPRC
C Tries to find a m/c for a component ; IH=comp.
pointer
C______________________________________________________________________

      SUBROUTINE STARCM(IH)
      INCLUDE SCM91.FOR
      DIMENSION KRANK(100)

C Identify comp no. and next op no.
      IA1=IATTRB(IH,1)
      IA2=IATTRB(IH,2)

C Calc set up code and op location
      NCELL=KOPCEL(NCOMOP(IA1)+IA2)
      NGRP=KOPGRP(NCOMOP(IA1)+IA2)

C Reorder queue now that new job has joined
      ISET=IFLOOR(NCELL,NGRP)
      NUM=ISIZOF(ISET)
      CALL QORDER(ISET,KRANK,NUM)

      DO 10 N=1,NUM
        JX=KRANK(NUM)
        NCOMP=IATTRB(JX,1)
        NOP=IATTRB(JX,2)
        IE=NCOMP*100+NOP

C Look for idle machine with this setup
        CALL CHKMCH(NCELL,NGRP,IE,JX)
10    CONTINUE

999   RETURN
      END

C______________________________________________________________________
C
C SEARCH FOR A MACHINE TO PROCESS A JOB
C______________________________________________________________________
```

```
      SUBROUTINE CHKMCH(IC,IG,IE,JX)
      INCLUDE SCM91.FOR
      DIMENSION IHOLD(3,20),NUM(3)

C Zero off array
      DO 10 I=1,3
         DO 5 J=1,KNOMCH(IC,IG)
            IHOLD(I,J)=0
5        CONTINUE
         NUM(I)=0
10    CONTINUE

C Order machines ; first machines with less than min.
C batch size

     DO 100 J=1,KNOMCH(IC,IG)
C Identify machine and current set up
        MNO=MCFIND(IC,IG,J)
        MACH=JMACH(MNO)
           NOWSET=IATTRB(JMACH(MNO),1)

C Only interested in idle machines, i.e.state=1
       IF(IATTRB(MACH,2).NE.1) GOTO 100

C FINDOP will return sequential set up no. if m/c is
C capable of IE
          CALL FINDOP(MNO,IE,0,IST)
          IF(IST.EQ.0) GOTO 100

          NMIN=KMCDT2(NMCOPS(MNO)+IATTRB(MACH,1))
          IF(NMIN.LE.0.OR.IATTRB(MACH,3).LT.NMIN) THEN
C Dedicated m/c or minimum no. batches has not been
C achieved
              NUM(1)=NUM(1)+1
              IHOLD(1,NUM(1))=MNO
          ELSEIF(NOWSET.EQ.IST) THEN
C This idle machine already has correct set up
              NUM(2)=NUM(2)+1
              IHOLD(2,NUM(2))=MNO
          ELSE
C Could change set up
              NUM(3)=NUM(3)+1
              IHOLD(3,NUM(3))=MNO
          ENDIF
100   CONTINUE

      DO 200 I=1,3
           IF(NUM(I).LE.0) GOTO 200
           DO 150 J=1,NUM(I)
              MNO=IHOLD(I,J)
              IF(I.LT.3) THEN
C Machine already has correct setup
                 CALL STPROC(JX,MNO)
              ELSE
C Need to change set up
                 CALL FINDOP(MNO,IE,0,ISET)
                 CALL ADSETQ(JMACH(MNO),ISET)
```

```
                  CALL SETUPS(MNO)
               ENDIF
C Check if this machine has accepted the job
               ISTATE=IATTRB(JMACH(IHOLD(I,J)),2)
               IF(ISTATE.NE.1) GOTO 999
150         CONTINUE
200   CONTINUE

999   RETURN
      END

C______________________________________________________________________
C
C   START MACHINING
C______________________________________________________________________

      SUBROUTINE STPROC(IH,MACHNO)
      INCLUDE SCM91.FOR

      MACH=JMACH(MACHNO)
      CALL MCNUMB(MACH,MCNUM1,I1,I2,I3)
      IST=IATTRB(MACH,1)

C If restart after breakdown or tool change, job will
C already be on machine
      IF(ISIZOF(IMACH(MACHNO)).GT.0) GOTO 10

         IF(IPOSOF(IH,IFLOOR(I1,I2)).EQ.0) THEN
             CALL BISUP(IRET,'Cannot find pallet')
             CALL INTRCT
         ELSE
            CALL REMOV(IH,IFLOOR(I1,I2))
         ENDIF

C Move identified pallet to m/c set
         CALL ADLAST(IH,IMACH(MACHNO))

C Function PROTIM checks remaining time, if any
10    TX=PROTIM(MACHNO,IST,IH)
      CALL CHBRTL(MACHNO,IST,TX)
      CALL SCHEDL(6,TX,IH)

C Show machine working
      ISTATE=2
      CALL MCHDSP(MACH,ISTATE)

C Update component state recorder
      ISTATE=2
      CALL UPDATC(IH,ISTATE)

      RETURN
      END

C______________________________________________________________________
C
C Find processing time
C______________________________________________________________________
```

```
      FUNCTION PROTIM(MACHNO,IST,IH)
      INCLUDE SCM91.FOR

C Schedule machining time from 3rd attribute if set
C (remaining time)
      IF(RATTRB(IH,3).GT.0.0) THEN
         PROTIM=RATTRB(IH,3)
      ELSE
C Get component and op. details
         IA1=KMCDT1(NMCOPS(MACHNO)+IST)/100
         IA2=KMCDT1(NMCOPS(MACHNO)+IST)-IA1*100
         PROTIM=ROPTIM(NCOMOP(IA1)+IA2)*FLOAT(KPLCAP
     &   (IA1))
         CALL RSETAT(IH,3,PROTIM)
      ENDIF

      RETURN
      END

C______________________________________________________
C
C Check if breakdown or tool change is due in current
C cycle
C______________________________________________________

      SUBROUTINE CHBRTL(MACHNO,IST,TX)
      INCLUDE SCM91.FOR

      TT=RTOLCH(NMCOPS(MACHNO)+IST)
      TB=RMCOP(MACHNO,1)
      IF(TT.LE.0.0.AND.RMTBF(MACHNO).LE.0.0) GOTO 999
C Check if breakdown or tool change is due in this
C machining time ;
C        if so , schedule whichever one is soonest
      IF(TT.LE.0.0.AND.RMTBF(MACHNO).GT.0.0) THEN
         IF(TB.LT.TX) CALL SCHEDL(4,TB,JMACH(MACHNO))
      ELSEIF(TB.LE.0.0) THEN
         IF(TT.LT.TX) CALL SCHEDL(7,TT,JMACH(MACHNO))
      ELSE
         IF(TB.LT.TT.AND.RMTBF(MACHNO).GT.0) THEN
           IF(TB.LT.TX) CALL SCHEDL(4,TB,JMACH(MACHNO))
         ELSE
           IF(TT.LT.TX) CALL SCHEDL(7,TT,JMACH(MACHNO))
         ENDIF
      ENDIF

 999  RETURN
      END


C______________________________________________________
C
C   END OF MACHINING
C______________________________________________________

      SUBROUTINE ENDPRC(NEXTM,MACHNO)
      INCLUDE SCM91.FOR
```

```
C Identify cell and group
      IA1=IATTRB(NEXTM,1)
      IA2=IATTRB(NEXTM,2)

      I1=KOPCEL(NCOMOP(IA1)+IA2)
      I2=KOPGRP(NCOMOP(IA1)+IA2)

C Find machine in group which holds NEXTM
      DO 10 I=1,KNOMCH(I1,I2)
         I3=I
         MACHNO=MCFIND(I1,I2,I3)
         IF(ISIZOF(IMACH(MACHNO)).EQ.0) GOTO 10
         IF(IHEDOF(IMACH(MACHNO)).EQ.NEXTM) GOTO 15
10    CONTINUE

      CALL BTEXTI('ENTITY NO ',INUMOF(1,NEXTM),6)
      CALL BISUP(IRET,'ERROR IN ENDPRC')
      CALL INTRCT

15    CONTINUE
      IH=NEXTM
      MACH=JMACH(MACHNO)
      IST=IATTRB(MACH,1)

C Increment batch counters
      CALL IINCAT(MACH,3,1)
      KBTREC(NMCOPS(MACHNO)+IST)= KBTREC
     &                        (NMCOPS(MACHNO)+IST)+1

C Decrement 'time to next breakdown' and 'tool change'
C recorders
      RMCOP(MACHNO,1)=RMCOP(MACHNO,1)-1.*RATTRB(IH,3)
      IF(RMCOP(MACHNO,1).LE.0.0) RMCOP(MACHNO,1)=0.1

      RTOLCH(NMCOPS(MACHNO)+IST)= RTOLCH (NMCOPS
     & (MACHNO)+IST) -RATTRB(IH,3)
      IF(RTOLCH(NMCOPS(MACHNO)+IST).LE.0.0)
     & RTOLCH(NMCOPS(MACHNO)+IST)=0.1

C Set remaining time counter to zero
      CALL RSETAT(IH,3,0.0)

      CALL BEHEAD(IMACH(MACHNO))


C Machine to idle
      ISTATE=1
      CALL MCHDSP(MACH,ISTATE)

C Update component recorder
      ISTATE=1
      CALL UPDATC(IH,ISTATE)

      RETURN
      END

C_____________________________________________________________
C
```

```
C Consequential event to end processing
C Send pallet to destinations and try start job
C______________________________________________________________

      SUBROUTINE CONEND(IH,MACHNO)
      INCLUDE SCM91.FOR

      CALL MCNUMB(JMACH(MACHNO),MNO,I1,I2,I3)
C Set next operation number and send to dest.
      NCOMP=IATTRB(IH,1)
      LOP=IATTRB(IH,2)

C Increment production counter
      CALL INCRM(IH)

C Know current cell is I1, group is I2, m/c is I3
C Identify next cell, machine group, and operation
      NXOP=IATTRB(IH,2)

      IY1=KOPCEL(NCOMOP(NCOMP)+NXOP)
      IY2=KOPGRP(NCOMOP(NCOMP)+NXOP)
      IF((I1.NE.IY1).OR.(I2.NE.IY2)) THEN
        JENT=IH
        CALL STMOVE(IY1,IY2,JENT,LOP)
      ELSE
C Next operation is still in this group
        CALL ADLAST(IH,IFLOOR(IY1,IY2))
C Update wip counters
        KCMWIP(NCOMOP(NCOMP)+LOP)=KCMWIP(NCOMOP(NCOMP)
     &  +LOP)-1
        KCMWIP(NCOMOP(NCOMP)+NXOP)=KCMWIP(NCOMOP(NCOMP)
     &   +NXOP)+1

      ENDIF


C Try to find job for machine
      CALL STARMC(MACHNO)


      IF(IPOSOF(IH,IFLOOR(I1,I2)).GT.0) THEN
C Try to find machine for job
         CALL STARCM(IH)
      ENDIF

      RETURN
      END

C______________________________________________________________
C
C LOOK FOR A JOB FOR A MACHINE
C______________________________________________________________

      SUBROUTINE STARMC(MACHNO)
      INCLUDE SCM91.FOR

C Identify location
      MACH=JMACH(MACHNO)
```

```
      CALL MCNUMB(MACH,NMCH,IC,IG,IM)
      NUM=ISIZOF(IFLOOR(IC,IG))
      IF(NUM.EQ.0) GOTO 99

C Find set up number
      NSET=IATTRB(MACH,1)

C Compare no.batches completed on this set up with
C minimum
      NDONE=IATTRB(MACH,3)
      NMIN=KMCDT2(NMCOPS(MACHNO)+NSET)

      IF(NMIN.EQ.0.OR.NDONE.LT.NMIN) THEN
         CALL CHKMC2(MACHNO,NSET)
      ELSE
         CALL CHKMC3(MACHNO,NUM,NSET)
      ENDIF

99    RETURN
      END

C_______________________________________________________
C
C Find next job for dedicated machine
C_______________________________________________________

      SUBROUTINE CHKMC2(MACHNO,NSET)
      INCLUDE SCM91.FOR

      IA1=KMCDT1(NMCOPS(MACHNO)+NSET)/100
      IA2=KMCDT1(NMCOPS(MACHNO)+NSET)-IA1*100
      CALL EXISTS(MACHNO,1,IA1,IA2,IH,IEXIST)
      IF(IEXIST.EQ.1) CALL STPROC(IH,MACHNO)
      RETURN
      END
C_______________________________________________________
C
C   Find next job for a machine which can change set up
C_______________________________________________________

      SUBROUTINE CHKMC3(MACHNO,NUM,NSET)
      INCLUDE SCM91.FOR
      DIMENSION KRANK(100)

      MACH=JMACH(MACHNO)
      CALL MCNUMB(MACH,NMCH,IC,IG,IM)
      ISET=IFLOOR(IC,IG)
      CALL QORDER(ISET,KRANK,NUM)

      DO 10 N=1,NUM
         JX=KRANK(N)
         IA1=IATTRB(JX,1)
         IA2=IATTRB(JX,2)
         IE=100*IA1+IA2
         CALL FINDOP(MACHNO,IE,0,IST)
         IF(IST.EQ.0) GOTO 10

C Machine is capable of this op
```

```
         IF(NSET.EQ.IST) THEN
            CALL STPROC(JX,MACHNO)
            GOTO 99
         ELSEIF(LMCSET(MACH,IST).EQ.1) THEN
            CALL ADSETQ(MACH,IST)
            CALL SETUPS(MACHNO)
            GOTO 99
         ENDIF
10    CONTINUE
99    RETURN
      END


C_______________________________________________________
C
C   Move pallet entity to next operation or out of shop
C Input:  Next cell, next group, IH = pallet ID, LOP =
C last op no.
C_______________________________________________________

      SUBROUTINE STMOVE(NXTCL,NXTGP,IH,LOP)
      INCLUDE SCM91.FOR

      NCOMP=IATTRB(IH,1)
      NOP=IATTRB(IH,2)

      IF(NXTCL.EQ.0) THEN
C No more operations
         CALL ADLAST(IH,IWRLDQ)
         CALL RECORD(IH)
         CALL ZERPLT(IH)

C Update WIP counters
         CALL CHGWIP(-1,NCOMP,LOP)
         KCMREC(NCOMP,2)=KCMREC(NCOMP,2)+KPLCAP(NCOMP)

      ELSE
         CALL SCHMOV(IH,NXTCL,NXTGP,LOP)
C Update WIP counters
         KCMWIP(NCOMOP(NCOMP)+NOP)=KCMWIP(NCOMOP(NCOMP)
     &   +NOP)+1
         KCMWIP(NCOMOP(NCOMP)+LOP)=KCMWIP(NCOMOP(NCOMP)
     &   +LOP)-1
C Update component recorder
         ISTATE=5
         CALL UPDATC(IH,ISTATE)
      ENDIF

      RETURN
      END

C_______________________________________________________
C
C   Schedule only one pallet in dummy and set end move
C   time for other pallets
C   Input:  IH = new pallet to be added in dummy
C_______________________________________________________
```

```
      SUBROUTINE SCHMOV(IH,NC,NG,NOP)
      INCLUDE SCM91.FOR

      NCOMP=IATTRB(IH,1)
      TRVT=RTVTIM(NCOMOP(NCOMP)+NOP)
      IDUM=IDUMMY(NC,NG)
      NDUM=ISIZOF(IDUM)

C If the set is empty, then schedule incoming element
      IF(NDUM.LE.0) THEN
         CALL RSETAT(IH,3,0.0)
         CALL SCHEDL(2,TRVT,IH)
         GOTO 999
      ENDIF

C Identify which element is scheduled
      DO 100 N1=1,NDUM
         IH1=IDNTOF(IDUM,N1)
         T3=TIMECL(IH1)
         IF(T3.GT.0.0.AND.RATTRB(IH1,3).EQ.0.0) GOTO
200
100   CONTINUE

      CALL BTEXTI('ERROR IN SCHMOVE',NC*100+NG,5)
      CALL BISUP(IRET,'>')
      CALL INTRCT

C If new element will have finished moving before
C currently scheduled element, then make new element
C the scheduled element
C If not, then set att 3 to arrival time
200   T1=STIME(1)
      T2=T1+TRVT
      IF(T2.GE.T3) THEN
C New element finishes moving AFTER scheduled element
         CALL RSETAT(IH,3,T2)
      ELSE
C New element finishes moving BEFORE currently
C scheduled element
         CALL DESCHD(IH1)
         CALL RSETAT(IH1,3,T3)
         CALL RSETAT(IH,3,0.0)
         CALL SCHEDL(2,TRVT,IH)
      ENDIF

999   CALL ADLAST(IH,IDUM)

      RETURN
      END

C_____________________________________________________
C
C CHECK WHETHER IT IS WORTHWHILE TO SET UP THE MACHINE
C  IN:  MACH = machine entity pointer, IOP = serial set
C  up number
C_____________________________________________________

      FUNCTION LMCSET(MACH,IOP)
```

```
      INCLUDE SCM91.FOR

      LMCSET=0
      CALL MCNUMB(MACH,MACHNO,IC,IG,IM)
      TSET=RMCDT1(NMCOPS(MACHNO)+IOP)
      NCODE=KMCDT1(NMCOPS(MACHNO)+IOP)

C Find if other machines in the group are already set
C up for this op and working
      NUM=0
      DO 10 N=1,KNOMCH(IC,IG)
         MNO=MCFIND(IC,IG,N)
         IF(KMCDT1(NMCOPS(MNO)+IATTRB(JMACH(MNO),1))
     &   .EQ.NCODE) THEN
C Machine is capable of this op and is now set up or
being set up for this op
               IF(ISIZOF(IMACH(MNO)).GT.0) THEN
                  JX=IHEDOF(IMACH(MNO))

                  IF(IEVNTF(JX).GT.0) THEN
C Machine is working
                     TREM=TIMECL(JX)-STIME(1)
                  ELSEIF(IEVNTF(JMACH(MNO)).EQ.11) THEN
C Include a machine that is currently tool changing
                     TREM=TIMECL(JMACH(MNO))-STIME(1)
                     TREM=TREM+RATTRB(JX,3)
                  ENDIF

                  IF(TREM.LT.TSET) NUM=NUM+1
               ELSE
C If machine is being set up for this job, then count
it in
                  IS=IATTRB(JMACH(MNO),2)
                  IF(IS.EQ.1.OR.IS.EQ.3.OR.IS.EQ.6)
     &            NUM=NUM+1
               ENDIF
         ENDIF
10    CONTINUE

C NUM is the number of pallets which can be machined on
C this setup and which will finish before another
C machine could be set up.  Find how many pallets are
C waiting in the floor set for this set up. If it is
C greater than NUM, then set up another one.  Batched,
C group and assembly ops must take account of no. sets
C available

      IA1=KMCDT1(NMCOPS(MACHNO)+IOP)/100
      IA2=KMCDT1(NMCOPS(MACHNO)+IOP)-IA1*100
      CALL EXISTS(MACHNO,NUM+1,IA1,IA2,IH,IEXIST)

      IF(IEXIST.EQ.1) THEN
         LMCSET=1
      ENDIF
      RETURN
      END

C_______________________________________________________________
```

```
C
C  Add machine to list of jobs for setting up
C______________________________________________________

      SUBROUTINE ADSETQ(MACH,IOP)
      INCLUDE SCM91.FOR

      CALL ADLAST(MACH,ISETQ)
      CALL ISETAT(MACH,1,IOP)

      RETURN
      END
C______________________________________________________
C
C Start setup
C______________________________________________________

      SUBROUTINE SETUPS(MNO)
      INCLUDE SCM91.FOR

      IF(MNO.EQ.0) THEN
         I1=1
         I2=ISIZOF(ISETQ)
         NUM=1
      ELSE
         I1=IPOSOF(JMACH(MNO),ISETQ)
         I2=I1
         NUM=I1
      ENDIF

      IF(ISIZOF(ISETQ).EQ.0) GOTO 999
      DO 10 J=I1,I2
         MACH=IDNTOF(ISETQ,NUM)
         IOP=IATTRB(MACH,1)
         CALL MCNUMB(MACH,MACHNO,IC,IG,IM)

         CALL REMOV(MACH,ISETQ)
         ISTATE=3
         CALL MCHDSP(MACH,ISTATE)

C Schedule end of set up
            IF(RMCOP(MACHNO,2).GT.0.0) THEN
               TSET=RMCOP(MACHNO,2)
               RMCOP(MACHNO,2)=0.
            ELSE
               TSET=RMCDT1(NMCOPS(MACHNO)+IOP)
            ENDIF
            CALL SCHEDL(3,TSET,MACH)

10    CONTINUE
999   RETURN
      END

C______________________________________________________
C
C   COMPLETION OF SET-UP
C______________________________________________________
```

```
      SUBROUTINE SETUPC(MACH)
      INCLUDE SCM91.FOR

C Zero off batch counter on this set up
      CALL ISETAT(MACH,3,0)

      CALL MCNUMB(MACH,MACHNO,IC,IG,IM)

C New set up no.
      ISU=IATTRB(MACH,1)

C Increment no. set ups recorder
      KSUREC(NMCOPS(MACHNO)+ISU)=KSUREC(NMCOPS(MACHNO)
     &+ISU)+1

C Record idle state
      ISTATE=1
      CALL MCHDSP(MACH,ISTATE)

      RETURN
      END


C_______________________________________________________________
C
C Routine to increment op. code attribute
C INPUT:IH=pointer of entity
C_______________________________________________________________

      SUBROUTINE INCRM(IH)
      INCLUDE SCM91.FOR
      DIMENSION IA3(MXALT+1)

C Zero off array
      DO 20 N=1,MXALT+1
        IA3(N)=0
20    CONTINUE

C Get comp. no. and current op. code
      IA1=IATTRB(IH,1)
      IA2=IATTRB(IH,2)

C Determine skip (if any)
      CALL IINCAT(IH,2,1+KNOSKP(NCOMOP(IA1)+IA2))

C Determine if alt. routes exist
      IA3(1)=IATTRB(IH,2)
      NO=1
      DO 10 I=1,MXALT
         IF(KALTOP(NCOMOP(IA1)+IA3(1),I).GT.0) THEN
           NO=NO+1
           IA3(NO)=KALTOP(NCOMOP(IA1)+IA3(1),I)-IA1*100
         ENDIF
10    CONTINUE

      IF(NO.GT.1) THEN
          CALL CHSALT(IA1,NO,IA3,IWHICH)
          CALL ISETAT(IH,2,IA3(IWHICH))
```

```
          CALL ISETAT(IH,7,IWHICH)
      ENDIF

      RETURN
      END
C_____________________________________________________________________
C
C Choose route from alternates
C_____________________________________________________________________

      SUBROUTINE CHSALT(NCOMP,NO,IA3,IWHICH)
      INCLUDE SCM91.FOR
      DIMENSION IA3(MXALT+1),INM(MXALT+1),
     &IMUCH(MXALT+1), PJOBS(MXCMP),RAVE(MXALT+1),
     &RDIFF(4)

      DATA RDIFF/270.,330.,190.,60./

C Zero arrays
      DO 1 N=1,MXALT+1
        INM(N)=0
        IMUCH(N)=0
1     CONTINUE

C Find first op in branch
      NFOP=KFOPBR(NCOMOP(NCOMP)+IA3(1))

      GOTO (100,200,200,200,500,500,500,800,500,500)
     & NRULE(2)

C 1. Ratio of pallets according to preferred route
C or bottleneck op times
100   RMIN=10.**5.
      DO 50 N=1,NO
        NOP=IA3(N)
        INDX=NCOMOP(NCOMP)+NOP
C Find least achieved decision proportion
        RTEMP=FLOAT(NRTOUT(INDX))/RATIO(INDX)
        IF(RTEMP.LT.RMIN) THEN
          RMIN=RTEMP
          IWHICH=N
        ENDIF
50    CONTINUE
C Increment decision reocrders
      INDX=NCOMOP(NCOMP)+IA3(IWHICH)
      NRTOUT(INDX)=NRTOUT(INDX)+1
      GOTO 99

C     2.   Total no. jobs in partial route

200   DO 20 N=1,NO
        NOP=IA3(N)
        NSTG=10
        RTOT=0.
        CALL RTLOAD(NCOMP,NOP,NSTG,NFOP,PJOBS)
        INM(N)=N
        DO 21 K = 1,MXCMP
          RTOT=RTOT+PJOBS(K)
```

```
21        CONTINUE

          IF(NRULE(2).EQ.2) THEN
            IMUCH(N)=IFIX(RTOT)
          ENDIF
20      CONTINUE

        CALL SORT(NO,INM,IMUCH)
        IWHICH=INM(1)
        GOTO 99

C     Work before bottleneck
C     5.  Number of jobs
C     6.  Workload for bottleneck
C     7.  Earliest expected start for decision job

500     DO 70 N=1,NO
          NOP=IA3(N)
C Locate which stage of route is the bottleneck
          CALL BTLNK1(NCOMP,NOP,NFOP,IBTLNK,MCB)
C Count no. jobs and workload upto and including
C bottleneck
          CALL RTLOAD(NCOMP,NOP,IBTLNK,NFOP,PJOBS)

          INM(N)=N
          RTOT=0.
          DO 71 K=1,MXCMP
            RTOT=RTOT+PJOBS(K)
71        CONTINUE
C Calculate the workload of these jobs at the
C bottleneck
          RLOAD=0.0
          DO 72 K=1,MXMCOP(MCB)
            IE=KMCDT1(NMCOPS(MCB)+K)
            NC=IE/100
            NOPN=IE-NC*100
            RLOAD=RLOAD+PJOBS(NC)*ROPTIM(NCOMOP(NC)+NOPN)
     &      *KPLCAP(NC)
72        CONTINUE

          CALL STATUS(MCB,XMUCH4,XMUCH5)
          IF(NRULE(2).EQ.5) THEN
            IMUCH(N)=IFIX(RTOT)
          ELSEIF(NRULE(2).EQ.6) THEN
            IMUCH(N)=IFIX(RLOAD)
          ELSEIF(NRULE(2).EQ.7) THEN
            IMUCH(N)=IFIX(RLOAD+XMUCH4+XMUCH5)
          ELSEIF(NRULE(2).EQ.9) THEN
            IMUCH(N)=IFIX(RLOAD+XMUCH4)
          ELSE
            IMUCH(N)=IFIX(RLOAD+XMUCH5)
          ENDIF

70      CONTINUE

        CALL SORT(NO,INM,IMUCH)
        IWHICH=INM(1)
```

```
      GOTO 99

C    8.  Choose route with least average flowtime
800    RMIN=10.**5.
       DO 10 I=1,NO
         IF(NCMOUT(NCOMP,I).LE.0)  THEN
           RAVE(I)=0.
         ELSE
           RAVE(I)=CMFLW(NCOMP,I)/FLOAT(NCMOUT(NCOMP,I))
           IF(I.EQ.2)  RAVE(I)=RAVE(I)-RDIFF(NCOMP)
         ENDIF
         IF(RAVE(I).LT.RMIN)  THEN
           IWHICH=I
           RMIN=RAVE(I)
         ENDIF
10     CONTINUE

99     RETURN
       END
C______________________________________________________________
C
C  Find number of jobs at each stage of this partial
route
C______________________________________________________________

          SUBROUTINE RTLOAD(NCOMP,NOP,NSTG,NFOP,PJOBS)
          INCLUDE SCM91.FOR
          DIMENSION PJOBS(MXCMP),RJOBS(MXCMP)

          DO 1 K=1,MXCMP
            PJOBS(K)=0.0
1         CONTINUE

          DO 30 I=1,NSTG
            INDX=NCOMOP(NCOMP)+NOP
            IF(KFOPBR(INDX).NE.NFOP)  GOTO 31
C Take into account jobs on machine, on floor and in
C transit
            IC=KOPCEL(INDX)
            IG=KOPGRP(INDX)
            MNO=MCFIND(IC,IG,1)
            CALL MCHLOD(MNO,RJOBS)
            DO 35 K=1,MXCMP
              PJOBS(K)=PJOBS(K)+RJOBS(K)
35          CONTINUE
            NOP=NOP+1+KNOSKP(INDX)
30        CONTINUE

31     RETURN
       END
C______________________________________________________________
C
C   Locate bottleneck on this partial route
C______________________________________________________________

          SUBROUTINE BTLNK1(NCOMP,NOP,NFOP,IBTLNK,MCB)
          INCLUDE SCM91.FOR
          DIMENSION KVAL(10),KPOS(10),MCBT(10)
```

```
      DO 1 N=1,10
        KVAL(N)=0
        KPOS(N)=0
1     CONTINUE

      IBTLNK=0
C Determine which stage is the bottleneck for this
C partial route
      NO1=0
      NO2=NOP
      DO 10 I=1,10
        INDX=NCOMOP(NCOMP)+NO2
        IF(KFOPBR(INDX).NE.NFOP) GOTO 11
        NO1=NO1+1
        IC=KOPCEL(INDX)
        IG=KOPGRP(INDX)
        MNO=MCFIND(IC,IG,1)
C Determine potential load on this machine per period
        RL=0.0
        DO 20 N=1,MXMCOP(MNO)
          IE=KMCDT1(NMCOPS(MNO)+N)
          NC=IE/100
          NO=IE-NC*100
          RL=RL+FLOAT(IATTRB(JORDR,NC))*ROPTIM(NCOMOP
     &     (NC)+NO)
20      CONTINUE
        KVAL(NO1)=IFIX(RL)
        KPOS(NO1)=I
        MCBT(NO1)=MNO
        NO2=NO2+1+KNOSKP(INDX)
10    CONTINUE

11    IF(NO1.GT.1) CALL SORT(NO1,KPOS,KVAL)
C SORT sorts them to least first; NO1 position contains
C most
      IBTLNK=KPOS(NO1)
      MCB=MCBT(IBTLNK)

      RETURN
      END
C_______________________________________________________
C
C Take job in progress and any breakdown into account
C_______________________________________________________

      SUBROUTINE STATUS(MNO,XMUCH2,XMUCH5)
      INCLUDE SCM91.FOR

      XMUCH2=0.0
      XMUCH5=0.0

      IF(ISIZOF(IMACH(MNO)).EQ.0) GOTO 99
      JX=IHEDOF(IMACH(MNO))
      NC=IATTRB(JX,1)
      NOPN=IATTRB(JX,2)
      TPROC=ROPTIM(NCOMOP(NC)+NOPN)*FLOAT(KPLCAP(NC))
```

```
      IF(IATTRB(JMACH(MNO),2).EQ.2) THEN
C Total job time has already been counted in, so find
C work already done to be able to deduct it for more
C accurate forecast
        XMUCH2=TIMECL(JX)-STIME(1)-TPROC
      ELSEIF(IATTRB(JMACH(MNO),2).EQ.4) THEN
C Add time to repair plus remaining machining time
        XMUCH5=TIMECL(JMACH(MNO))-STIME(1)
        XMUCH2=RATTRB(JX,3)-TPROC
      ENDIF

99    RETURN
      END
C______________________________________________________
C
C Determines current workload of a m/c
C Assumes same op. code does not appear on two
different
C set-up codes of a m/c
C INPUT:MACHNO=m/c no.
C______________________________________________________

      SUBROUTINE MCHLOD(MNO,RJOBS)
      INCLUDE SCM91.FOR
      DIMENSION RJOBS(MXCMP)

      DO 1 K=1,MXCMP
        RJOBS(K)=0.
1     CONTINUE
C Examine each op of a machine
      DO 10 N=1,MXMCOP(MNO)
         IE=KMCDT1(NMCOPS(MNO)+N)
         NCOMP=IE/100
         NOP=IE-NCOMP*100
C Wip recorder includes those in floor and those
C expected (in DUMMY) and any job on the machine
         NJOBS=KCMWIP(NCOMOP(NCOMP)+NOP)

C Find out how many other m/c in this group could do
C this op
         CALL MCNUMB(JMACH(MNO),MACHNO,IC,IG,IM)
         NOM=0
         DO 20 M=1,KNOMCH(IC,IG)
           NMC=MCFIND(IC,IG,M)
C Can this machine do this op
           CALL FINDOP(NMC,IE,0,IST)
           IF(IST.GT.0) NOM=NOM+1
20       CONTINUE
C NOM should always be at least 1
         SJOBS=FLOAT(NJOBS)/FLOAT(NOM)
         RJOBS(NCOMP)=RJOBS(NCOMP)+SJOBS

10    CONTINUE

      RETURN
      END

C______________________________________________________
```

```
C
C Find a m/c and set up which requires a particular
C comp+op.
C IND=0 means any set up,=1 means current set up of
that
C m/C
C_______________________________________________________

      SUBROUTINE FINDMC(IA1,IA2,IND,MNO,IST)
      INCLUDE SCM91.FOR

      IC=KOPCEL(NCOMOP(IA1)+IA2)
      IG=KOPGRP(NCOMOP(IA1)+IA2)
      IE=IA1*100+IA2
      DO 5 J=1,KNOMCH(IC,IG)
         MNO=MCFIND(IC,IG,J)
         IF(IND.EQ.0) THEN
            IOP=0
         ELSE
            IOP=IATTRB(JMACH(MNO),1)
         ENDIF
         CALL FINDOP(MNO,IE,IOP,IST)
         IF(IST.GT.0) GOTO 999
5     CONTINUE

999   RETURN
      END

C_______________________________________________________
C
C Test if a particular op. is part of a machine setup
C INPUT:MACHNO=m/c no.;IE= set up code being
C searched,IOP=0,if for any
C set up,>0 if for a particular set up; IOP=seq set up
C no.
C OUTPUT:IST= seq set up no.
C_______________________________________________________

      SUBROUTINE FINDOP(MACHNO,IE,IOP,IST)
      INCLUDE SCM91.FOR

      IST=0
      IF(MXMCOP(MACHNO).EQ.0) GOTO 999
      IF(IOP.EQ.0) THEN
         IS=1
         IF=MXMCOP(MACHNO)
      ELSE
         IS=IOP
         IF=IOP
      ENDIF

C IS = start point, IF = finish point
      DO 10 J=IS,IF
            IF(KMCDT1(NMCOPS(MACHNO)+J).EQ.0) GOTO 10
            IF(KMCDT1(NMCOPS(MACHNO)+J).EQ.IE) THEN
               IST=J
               GOTO 999
            ENDIF
```

```
10     CONTINUE

999    RETURN
       END

C_____________________________________________________________
C
C RETURNS PREVIOUS OP. NO.
C INPUT:IA1=comp. no.,IA2=current op. no.
C OUTPUT:IA3=prev. op. no.(list: there may be
C alternatives)
C NO=no. of possible alternatives for prev. op.
C_____________________________________________________________

       SUBROUTINE PREVOP(IA1,IA2,IA3)
       INCLUDE SCM91.FOR
       DIMENSION IA3(MXALT+1)

C  Zero off array to hold alternate op. routes
       DO 20 N=1,MXALT+1
         IA3(N)=0
20     CONTINUE

       NO=1
       DO 10 I=IA2-1,1,-1
            IX=I+1+KNOSKP(NCOMOP(IA1)+I)
            IF(IX.EQ.IA2) THEN
C Found the place;find list of alternatives,if any
               IA3(1)=I
               DO 5 J=1,MXALT
                  IF(KALTOP(NCOMOP(IA1)+I,J).GT.0) THEN
                     NO=NO+1
                     IA3(NO)=KALTOP(NCOMOP(IA1)+I,J)-IA1
     &               *100
                  ENDIF
5              CONTINUE
               RETURN
            ENDIF
10     CONTINUE

       CALL BISUP(IRET,'ERROR IN PREVOP')
       CALL INTRCT
       END
C_____________________________________________________________
C
C     START TOOL CHANGE
C_____________________________________________________________

       SUBROUTINE TLCHGS(MACH)
       INCLUDE SCM91.FOR

       CALL MCNUMB(MACH,MACHNO,IC,IG,IM)
       IF(RMCOP(MACHNO,2).GT.0.) THEN
C Tool change started in last shift
          TX=RMCOP(MACHNO,2)
          RMCOP(MACHNO,2)=0.
       ELSE
C New tool change:
```

```
C Update working time recorders
          JX=IHEDOF(IMACH(MACHNO))
          TREM=TIMECL(JX)-STIME(1)
          IF(TREM.EQ.0.0) TREM=0.01
          TDONE=RATTRB(JX,3)-TREM
          CALL DESCHD(JX)
          CALL RSETAT(JX,3,TREM)
C Update breakdown recorder
          RMCOP(MACHNO,1)=RMCOP(MACHNO,1) -1.*TDONE
          IF(RMCOP(MACHNO,1).LE.0.0) RMCOP(MACHNO,1)=0.1
C Schedule end of tool change
          NSET=IATTRB(JMACH(MACHNO),1)
          TX=RTLCHG(NMCOPS(MACHNO)+NSET)
      ENDIF

      CALL SCHEDL(8,TX,MACH)
      NSTATE=5
      CALL MCHDSP(MACH,NSTATE)

C Update component recorder
      ISTATE=4
      CALL UPDATC(JX,ISTATE)

      RETURN
      END
C______________________________________________________________________
C
C  END OF TOOL CHANGE
C______________________________________________________________________

      SUBROUTINE TLCHGF(MACH,IH,MACHNO)
      INCLUDE SCM91.FOR

      CALL MCNUMB(MACH,MACHNO,IC,IG,IM)

C Reset time counter for next tool change
      NSET=IATTRB(MACH,1)
      NCOMP=KMCDT1(NMCOPS(MACHNO)+NSET)/100
      NOP=KMCDT1(NMCOPS(MACHNO)+NSET)-NCOMP*100
      TNEXT=FLOAT(KMCDT3(NMCOPS(MACHNO)+NSET))*
     &              ROPTIM(NCOMOP(NCOMP)+NOP)
      RTOLCH(NMCOPS(MACHNO)+NSET)=TNEXT

C Restart work on this machine
      IH=IHEDOF(IMACH(MACHNO))

      RETURN
      END
C______________________________________________________________________
C
C BREAKDOWN
C______________________________________________________________________
C
      SUBROUTINE BRDOWN(NEXTM)
      INCLUDE SCM91.FOR

C NEXTM is m/c identity
      CALL MCNUMB(NEXTM,MACHNO,IC,IG,IM)
```

```
      MACH=JMACH(MACHNO)
C Change machine display
      ISTATE=4
      CALL MCHDSP(MACH,ISTATE)

C Set attribute to remaining machining time
      JX=IHEDOF(IMACH(MACHNO))
      TREM=TIMECL(JX)-STIME(1)
      TDONE=RATTRB(JX,3)-TREM
      CALL RSETAT(JX,3,TREM)
C Update component recorder
      ISTATE=3
      CALL UPDATC(JX,ISTATE)

C Decrement time to next tool change recorder
      IST=IATTRB(MACH,1)
      RTOLCH(NMCOPS(MACHNO)+IST)=RTOLCH(NMCOPS(MACHNO)
     & +IST)-TDONE
      IF(RTOLCH(NMCOPS(MACHNO)+IST).LE.0.0) RTOLCH
     & (NMCOPS(MACHNO)+IST)=0.1

C Deschedule end of processing
      CALL DESCHD(JX)
C Find repair time
       TTR= SERLAN(100.,2,50+MACHNO)

C Schedule end of repair
      CALL SCHEDL(5,TTR,JMACH(MACHNO))

99    RETURN
      END
C_______________________________________________________________
C
C END OF BREAKDOWN
C_______________________________________________________________
C
      SUBROUTINE UPMACH(NEXTM)
      INCLUDE SCM91.FOR
C
C Find time to next breakdown
      CALL MCNUMB(NEXTM,MACHNO,IC,IG,IM)
      CALL RNDNUM(0,MACHNO,UPROB)
C MTBF is supplied in hours
      BKDNTM=-1.0*60.*RMTBF(MACHNO)*ALOG(UPROB)
      RMCOP(MACHNO,1)=BKDNTM

      RETURN
      END


C_______________________________________________________________
C
C  Record information from finished job
C_______________________________________________________________

      SUBROUTINE RECORD(IH)
      INCLUDE SCM91.FOR
```

```
      NCOMP=IATTRB(IH,1)
C Flow time recorders
      TLNCH=RATTRB(IH,4)
      FLOWT=STIME(1)-TLNCH

C Compare with max recorded
      IF(FLOWT.GT.FMXMN(NCOMP,1)) FMXMN(NCOMP,1)=FLOWT
C Compare with min recorded
      IF(FLOWT.LT.FMXMN(NCOMP,2)) FMXMN(NCOMP,2)=FLOWT

C By component and route
      NRT=IATTRB(IH,7)
      CMFLW(NCOMP,NRT)=CMFLW(NCOMP,NRT)+FLOWT
      NCMOUT(NCOMP,NRT)=NCMOUT(NCOMP,NRT)+1
      CMFLW(NCOMP,0)=CMFLW(NCOMP,0)+FLOWT
      NCMOUT(NCOMP,0)=NCMOUT(NCOMP,0)+1
      SIGSQ(NCOMP,0)=SIGSQ(NCOMP,0)+FLOWT**2.
      SIGSQ(NCOMP,NRT)=SIGSQ(NCOMP,NRT)+FLOWT**2.

      RETURN
      END

C_______________________________________________________
C
C Routine to test if a list of comp/op. exists in the
C specified qty.
C in a given set
C INPUT: NO=no. of comp./op. codes
C IA1=comp.,IA2=op. no.,IQ=qty.    (all three : list)
C OUTPUT:IH=pointers for entities found,IEXIST=no.
C found
C_______________________________________________________

      SUBROUTINE EXISTS(MACHNO,NSETS,IA1,IA2,IH,IEXIST)
      INCLUDE SCM91.FOR

      IEXIST=0
      IH=0
      LOTS=0

      CALL MCNUMB(JMACH(MACHNO),MNO,IC,IG,IM)
      ISET=IFLOOR(IC,IG)

C Normal machining of grouped operation
      IF(ISIZOF(ISET).EQ.0) GOTO 999
      IS=ISIZOF(ISET)
C For each comp./op.,check for existence of right qt.
      DO 20 J=1,IS
         IX=IDNTOF(ISET,J)
         IY1=IATTRB(IX,1)
         IY2=IATTRB(IX,2)
         IF(IY1.EQ.IA1.AND.IY2.EQ.IA2) THEN
C Found one
            IH=IX
            LOTS=LOTS+1
            IF(NSETS.EQ.1.OR.NSETS.EQ.LOTS) THEN
               IEXIST=1
               GOTO 999
```

```
            ENDIF
          ENDIF
20      CONTINUE


999     RETURN
        END
C______________________________________________________________________
C
C  Order the queue according to the despatching rule
C______________________________________________________________________

        SUBROUTINE QORDER(ISET,KRANK,NUM)
        INCLUDE SCM91.FOR

        DIMENSION KRANK(100),KVAL(100),KPOS(100),
       & KTEMP(100)

C Safety check
        IF(NUM.GT.100) THEN
          CALL BISUP(IRET,'QORDER > 100')
          CALL INTRCT
        ENDIF

C Zero off arrays
        DO 40 I=1,100
          KRANK(I)=0
          KVAL(I)=0
40      CONTINUE

        DO 10 N=1,NUM

            JX=IDNTOF(ISET,N)
            NCMP=IATTRB(JX,1)
            NOP=IATTRB(JX,2)

            GOTO(100,200,300,400,500,600) NRULE(1)

C Rule 1, First come, first served
100         KVAL(N)=N
            GOTO 10

C Rule 2, Shortest processing time
C N.B Destination (i.e. alternative route) is decided
C before dispatching rule operates
200         PROTIM=ROPTIM(NCOMOP(NCMP)+NOP)*FLOAT(KPLCAP
       &    (NCMP))
            KVAL(N)=IFIX(PROTIM)
            GOTO 10

C Rule 3, Achievement ratio
300         CALL CALCUL(NCMP,NOP,XMUCH,IMUCH)
            KVAL(N)=IFIX(XMUCH*100.)
            GOTO 10

C Rule 4, Urgency ratio
400         CALL CALCUL(NCMP,NOP,TMUCH,IMUCH)
            KVAL(N)=IMUCH
```

```
          GOTO 10

C Rule 5, Maximum remaining operations
500       NXTOP=NOP
          NREM=0
          DO 30 I=1,20
             IF(NXTOP.EQ.MXCMOP(NCMP))  THEN
               KVAL(N)=20-NREM
               GOTO 31
             ENDIF
             NXTOP=NXTOP+KNOSKP(NCOMOP(NCMP)+NXTOP)
             NREM=NREM+1
30        CONTINUE
31        GOTO 10

C Rule 6, Launch time
600       KVAL(N)=IFIX(RATTRB(JX,4)-RSTPD)


10     CONTINUE

C Put pointers into KTEMP array and position into KPOS
C array
       DO 700 N=1,NUM
          KPOS(N)=N
          KTEMP(N)=IDNTOF(ISET,N)
700    CONTINUE

C Sort values, most critical first
C  N.B In all cases, a low KVAL(N) value indicates
C higher priority
       IF(NUM.GT.1)  CALL SORT(NUM,KPOS,KVAL)

C Order values in KRANK array from sort results
       DO 20 N=1,NUM
         NPOS=KPOS(N)
         KRANK(N)=KTEMP(NPOS)
20     CONTINUE

       RETURN
       END
C_____________________________________________________
C
C Handling jobs which have just arrived
C_____________________________________________________

       SUBROUTINE ARRIVE(NEXTM)
       INCLUDE SCM91.FOR

C Identify component type
       NCMP=IATTRB(NEXTM,1)

C Remove from world
       IH=IHEDOF(IWRLDQ)
       CALL CHILD(IH,KLDNUM(NCMP))
       CALL CHDESC(IH,DESCR(NCMP))
       CALL ISETAT(IH,1,NCMP)
       CALL ISETAT(IH,2,1)
```

```
C Record launch time - used for flowtime and
C despatching rule
      CALL RSETAT(IH,4,STIME(1))

C Set component state and start time of current state
      CALL ISETAT(IH,5,1)
      CALL RSETAT(IH,6,STIME(1))

C Routing marker
      CALL ISETAT(IH,7,1)

C Add to floor set for first op
      CALL BEHEAD(IWRLDQ)
      NCL=KOPCEL(NCOMOP(NCMP)+1)
      NGP=KOPGRP(NCOMOP(NCMP)+1)
      CALL ADLAST(IH,IFLOOR(NCL,NGP))

C Increment WIP counters
      CALL CHGWIP(1,NCMP,1)

C Try to start job
      CALL STARCM(IH)

C Schedule next arrival
      CALL SCHEDL(1,SNEGX(RARRVL(NCMP),NRNS(NCMP)),
     &JCPT(NCMP))

      RETURN
      END
C_____________________________________________________
C_____________________________________________________
C  Change WIP recorder level
C_____________________________________________________

      SUBROUTINE CHGWIP(NCHG,NCOMP,NOP)
      INCLUDE SCM91.FOR

C Record current state of WIP and last time of change
      LEVL=KCMREC(NCOMP,1)
      RTIM=STIME(1)-RCMREC(NCOMP,1)
      RCMREC(NCOMP,1)=STIME(1)
C Calculate area and add to cumulative total
      AREA=RTIM*FLOAT(LEVL)
      RCMREC(NCOMP,2)=RCMREC(NCOMP,2)+AREA

C Amend WIP level
      KCMREC(NCOMP,1)=KCMREC(NCOMP,1)+NCHG
      KCMWIP(NCOMOP(NCOMP)+NOP)=KCMWIP(NCOMOP(NCOMP)
     &+NOP)+NCHG

C Test for min/max change
      IF(KCMREC(NCOMP,1).LT.KCMREC(NCOMP,5)) THEN
         KCMREC(NCOMP,5)=KCMREC(NCOMP,1)
      ELSEIF(KCMREC(NCOMP,1).GT.KCMREC(NCOMP,6)) THEN
         KCMREC(NCOMP,6)=KCMREC(NCOMP,1)
      ENDIF
```

```
      RETURN
      END

C___________________________________________________________
C___________________________________________________________
C CALCULATE WIP FOR SUBSEQUENT OPS
C Input:IA1=component IA2=op. no.;Output IMUCH=wip(in
C pallets)
C___________________________________________________________

      SUBROUTINE CALCUL(IA1,IA2,TMUCH,IMUCH)
      INCLUDE SCM91.FOR

C Want to find first op at end of the branch, if op in
C question is on a branch, or the next op if op in
C question is not on a branch

      JMUCH=0
C Period duration
      TDUR = RATTRB(JCONT,2)
C Travel time to next op in periods
      TLDTM=RTVTIM(NCOMOP(IA1)+IA2)/TDUR

      IF(KFOPBR(NCOMOP(IA1)+IA2).GT.0) THEN
C This is part of a branch
          IA31=KFOPBR(NCOMOP(IA1)+IA2)
          IA3=IA2
          DO 80 I=1,100
C Find op which is not on this branch i.e. where ITS
C first op in branch is different
              IF (KFOPBR(NCOMOP(IA1)+IA3).NE.IA31) GOTO
90
              IF(I.NE.1) THEN
C Sum WIP and lead time to end of branch
                 JMUCH=JMUCH+KCMWIP(NCOMOP(IA1)+IA3)
                 RT=ROPTIM(NCOMOP(IA1)+IA3)*FLOAT(KPLCAP
     &           (IA1))
                 TLDTM=TLDTM+(RT/TDUR)+(RTVTIM(NCOMOP(IA1)
     &           +IA4)/TDUR)
              ENDIF
              IA3=IA3+KNOSKP(NCOMOP(IA1)+IA3)+1
80        CONTINUE
      ELSE
C Not on a branch, so take next op
          IA3=IA2+KNOSKP(NCOMOP(IA1)+IA2)+1
      ENDIF


C Count all WIP at downstream operations, starting from
C IA3 just identified, including all WIP on subsequent
C branches, but ignoring remaining WIP on this branch
C (if any)
90    DO 100 I=IA3,MXCMOP(IA1)
          IF(KOPCEL(NCOMOP(IA1)+I).LE.0) GOTO 105
          JMUCH=JMUCH+KCMWIP(NCOMOP(IA1)+I)
100   CONTINUE

C Convert to no. components
```

```
105    RMUCH=FLOAT(JMUCH)*FLOAT(KPLCAP(IA1))

C Current achievement against target
       SMUCH=FLOAT(KCMREC(IA1,2))

C Daily requirement
       DAYREQ=FLOAT(IATTRB(JORDR,IA1))

C WIP expressed as proportion of requirement
       TMUCH=(RMUCH+SMUCH)/DAYREQ

C Calculate lead time for one pallet thro' all
C remaining ops
C Lead time comprises travel time and op time
C Express lead time in no. shifts/days
       IA4=IA3
       DO 110 I=1,100
          IF(IA4.GE.MXCMOP(IA1)) GOTO 120
          RT=ROPTIM(NCOMOP(IA1)+IA4)*FLOAT(KPLCAP(IA1))
          TLDTM=TLDTM+(RT/TDUR)+(RTVTIM(NCOMOP(IA1)+IA4)
      &   /TDUR)
          IA4=IA4+KNOSKP(NCOMOP(IA1)+IA4)+1
110    CONTINUE
120    CONTINUE

C Calculate XMUCH as ratio of achievement to lead time.
C For same achievement, long lead time is more urgent.
       IF(TLDTM.LE.0.0) THEN
C Avoid 0/0 divide error
         XMUCH=0.0
       ELSE
         XMUCH=(TMUCH/TLDTM)*100.0
       ENDIF

       IF(XMUCH.GT.32500.0) THEN
          IMUCH=32500
       ELSEIF(XMUCH.LT.-32500.0) THEN
          IMUCH=-32500
       ELSE
          IMUCH=(XMUCH*1.0)
       ENDIF

       RETURN
       END
```

Appendix 1.3

Overlay 2 - File reading and Initialisation

```
C______________________________________________________________
C
C Read machine details and set up sets and entities
C______________________________________________________________

      SUBROUTINE GETMCH
      INCLUDE SCM91.FOR
      CHARACTER*12 IGNORE
      CHARACTER*12 ERRFN
      INTEGER*4 KERRJ

      DIMENSION ISET(50),IMIN(50),RTIM(50),LCELL(5,2),
     &  IMAX(50),RTOL(50),ITYP(50)


      NOCLS=0
      NCOUNT=0
      NCT1=0
      NCT2=0

      DO 15 M=1,MXSHP
C For each shop

      ERRFN=MFIL
      CALL BTEXT(MFIL)

      OPEN(UNIT=6,FILE=MFIL,STATUS='OLD',IOSTAT=KERRJ,
     & ERR=99999)

      READ(6,*,IOSTAT=KERRJ,ERR=99999) IGNORE,
     &LCELL(M,1),LCELL(M,2)

      NOCLS=NOCLS+LCELL(M,1)
C For each cell
      DO 20 I=LCELL(M,2),(LCELL(M,2)+LCELL(M,1)-1)

          KSHPNM(I)=M
          KPGENM(I)=1

C Read machine details into array
          READ(6,*,IOSTAT=KERRJ,ERR=99999) CNAME(I),
     &    KNOGRP(I)

C For each machine group
          DO 30 J=1,KNOGRP(I)

             READ(6,*,IOSTAT=KERRJ,ERR=99999) IGNORE,NCL,
     &       NGP,KNOMCH(I,J),KXCORD(I,J),KYCORD(I,J),IMK

          CTYP(I,J)=IGNORE(1:4)

C For each machine in the group
          DO 998 K=1,KNOMCH(I,J)
             COUNT=NCOUNT+1
             READ(6,*,IOSTAT=KERRJ,ERR=99999) RMTBF
     &        (NCOUNT)
 998      CONTINUE
```

```
            DO 4001 NX=1,50
               ISET(NX)=0
               IMIN(NX)=0
               RTIM(NX)=0
               RTOL(NX)=0
               IMAX(NX)=0
               ITYP(NX)=0
4001        CONTINUE

            K=0
4000        K=K+1

C For each set up of the machines
            READ(6,*,IOSTAT=KERRJ,ERR=99999) ISET(K),
     &      IMIN(K),RTIM(K),IMAX(K),RTOL(K)


3999        IF (ISET(K).NE.-1) GOTO 4000
            ISET(K)=0
            NUM=0

C Store machine set up data
            DO 307 NN=1,KNOMCH(I,J)
               NCT1=NCT1+1
               MCOPS=0
               IF(IMK.EQ.1) NUM=0
               DO 309 L=1,MXSET
                  NUM=NUM+1
                  IF(ISET(NUM).LE.0) GOTO 308
                  NCT2=NCT2+1
                  KMCDT1(NCT2)=ISET(NUM)
                  KMCDT2(NCT2)=IMIN(NUM)
                  RMCDT1(NCT2)=RTIM(NUM)
                  KMCDT3(NCT2)=IMAX(NUM)
                  RTLCHG(NCT2)=RTOL(NUM)

                  MCOPS=MCOPS+1

311               CONTINUE
309            CONTINUE
308            MXMCOP(NCT1)=MCOPS
307         CONTINUE

30       CONTINUE

20    CONTINUE
15    CONTINUE
      CLOSE(6)

      NOMCH=NCOUNT

      NMCOPS(1)=0
      DO 185 I=2,NOMCH
         NMCOPS(I)=NMCOPS(I-1)+MXMCOP(I-1)
185   CONTINUE

      RETURN
C--- handle error in input data file
```

```
99999 CALL FERROR(KERRJ,ERRFN)
      END


C_______________________________________________________________________
C
C Read component details
C_______________________________________________________________________
      SUBROUTINE GETOPN
      INCLUDE SCM91.FOR
      CHARACTER*10 IGNORE
      CHARACTER*12 ERRFN
      CHARACTER*15 OPNAME
      INTEGER*4 KERRJ

C---ENTITY DEFINITIONS

C Order control entity
      CALL DEFENT(JORDR,'ORDR',0)

C---CONTROL ENTITY---PARTS
      ERRFN=OFIL
      CALL BTEXT(OFIL)
      OPEN(UNIT=6,FILE=OFIL,STATUS='OLD',IOSTAT=KERRJ,
     &ERR=99999)
      J=0

      DO 62 I=1,NOCMP
C For each component

         READ(6,*,IOSTAT=KERRJ,ERR=99999) IGNORE,
     &   DESCR(I),KCOMP,KCLLNO(I),IREQT,KPLCAP(I),
     &   KCMREC(I,3),KCMREC(I,4),KLDNUM(I),RARRVL(I),
     &   NRNS(I)

          CALL ISETAT(JORDR,I,IREQT)


          J1=0
71        J=J+1
          J1=J1+1

C For each operation
         READ(6,*,IOSTAT=KERRJ,ERR=99999) OPNAME,
     &   KOPCEL(J),KOPGRP(J),NOSET,ROPTIM(J),RTVTIM(J),
     &   KNOSKP(J),KFOPBR(J),RATIO(J),(KALTOP(J,MM),
     &   MM=1,3)

          IF (KOPCEL(J).NE.-1) GOTO 71
          KOPCEL(J)=0
          MXCMOP(I)=J1

62     CONTINUE
       CLOSE(UNIT=6)

       NCOMOP(1)=0
       DO 185 I=2,NOCMP
```

```
         NCOMOP(I)=NCOMOP(I-1)+MXCMOP(I-1)
185   CONTINUE

      RETURN
C--- handle error in input data file
99999 CALL FERROR(KERRJ,ERRFN)
      END

C______________________________________________________________
C
C  FILE ERROR HANDLING ROUTINE
C______________________________________________________________

      SUBROUTINE FERROR(KERRJ,ERRFN)
      INCLUDE SCM91.FOR
      INTEGER*4 KERRJ
      CHARACTER*12 ERRFN
      CHARACTER*39 EMESS
      EMESS='UNIDENTIFIED ERROR'
      CALL BTEXT('ERROR IN INPUT FILE '//ERRFN)
      KCODE=INT2(MOD(KERRJ,256))
      IF (KCODE.EQ.20) EMESS='FILE DOES NOT EXIST'
      IF (KCODE.EQ.32) EMESS='INVALID DATA IN THIS
     & FILE'
      IF (KCODE.EQ.33) EMESS='INVALID NUMERIC DATA IN
     & THIS FILE'
      IF (KCODE.EQ.31) EMESS='INVALID CHARACTER DATA IN
     & THIS FILE'
      IF (KCODE.EQ.30) EMESS='INVALID LOGICAL DATA IN
     & THIS FILE'
      IF (KCODE.EQ.65) EMESS='SPECIFIED PATH NOT FOUND'
      IF (KCODE.EQ.58) EMESS='INVALID FILE NAME'
      IF (KCODE.EQ.57) EMESS='CANNOT OPEN THIS FILE'
      IF (KCODE.EQ.8) EMESS='INTEGER VALUE IS TOO BIG'
      IF (KCODE.EQ.1) EMESS='SYSTEM HAS RUN OUT OF
     & MEMORY'
      IF (KCODE.EQ.14) EMESS='ATTEMPTED TO READ PAST
     & END
     & OF FILE'
      IF (KCODE.EQ.13) EMESS='ATTEMPTED TO READ PAST
     & END OF RECORD'
      IF (KCODE.EQ.15) EMESS='SYSTEM I/O ERROR'
       CALL BTEXT(EMESS)
       CALL BISUP(KDUMMY,'HIT ENTER TO KILL PROGRAM')
       CALL KILL
       RETURN
       END
C______________________________________________________________
C
C READ INITIALISATION FILE
C______________________________________________________________

      SUBROUTINE GETBAJ
      INCLUDE SCM91.FOR
      CHARACTER*12 WRET
      INTEGER*4 KERRJ
      CALL BASUP(WRET,12,'Which init file?')
      DFIL=WRET
```

```
      OPEN (UNIT=6,FILE=WRET,STATUS='OLD',
     &  IOSTAT=KERRJ,ERR=99999)
      READ(6,*,IOSTAT=KERRJ,ERR=99999) NOCMP
      READ(6,*,IOSTAT=KERRJ,ERR=99999) MXSHP
      READ(6,*,IOSTAT=KERRJ,ERR=99999) NPLTS
      READ(6,*,IOSTAT=KERRJ,ERR=99999) NOENTS
      READ(6,*,IOSTAT=KERRJ,ERR=99999) NOATTR
      READ(6,*,IOSTAT=KERRJ,ERR=99999) NOSETS
      READ(6,*,IOSTAT=KERRJ,ERR=99999) NOMEMS
      READ(6,*,IOSTAT=KERRJ,ERR=99999) ISBLKL
      READ(6,*,IOSTAT=KERRJ,ERR=99999) MAXELS
      READ(6,*,IOSTAT=KERRJ,ERR=99999) WFIL
      READ(6,*,IOSTAT=KERRJ,ERR=99999) OFIL
      READ(6,*,IOSTAT=KERRJ,ERR=99999) MFIL
      READ(6,*,IOSTAT=KERRJ,ERR=99999) SFIL
      CLOSE(6)
      RETURN
99999 CALL FERROR(KERRJ,WRET)
      RETURN
      END

C______________________________________________________________
C  END OF SIMULATION
C______________________________________________________________
      SUBROUTINE ENDSIM
      INCLUDE SCM91.FOR
      CALL KILL
      RETURN
      END
C______________________________________________________________
C
C   READ WIP FILE
C______________________________________________________________

      SUBROUTINE PREFIL
      INCLUDE SCM91.FOR
C
C Prefill shop
      IF(WFIL.EQ.'NONE') RETURN
C Format: Comp no., Next opn, Qty of pallets waiting

      CALL BTEXT(WFIL)
      OPEN(UNIT=5,FILE=WFIL)
10    CONTINUE
         READ(5,*) NCOMP,NOP,NQTY

C -1 means end if this component
         IF (NCOMP.EQ.-1) GOTO 10

C -9 means end of file
         IF (NCOMP.EQ.-9) GOTO 99

C Add this quantity of components to each group's floor
C store

         NCEL=KOPCEL(NCOMOP(NCOMP)+NOP)
         NGRP=KOPGRP(NCOMOP(NCOMP)+NOP)
```

```
         DO 20 M=1,NQTY
            IX=IHEDOF(IWRLDQ)
            CALL ISETAT(IX,1,NCOMP)
            CALL ISETAT(IX,2,NOP)
            CALL CHILD(IX,KLDNUM(NCOMP))
            CALL CHDESC(IX,DESCR(NCOMP))
            CALL BEHEAD(IWRLDQ)
            CALL ADLAST(IX,IFLOOR(NCEL,NGRP))
C Increment total no. in system and no. at the op.
               KCMREC(NCOMP,1)=KCMREC(NCOMP,1)+1
               KCMWIP(NCOMOP(NCOMP)+NOP)=KCMWIP(NCOMOP
     &         (NCOMP)+NOP)+1
20          CONTINUE

      GOTO 10

99    CONTINUE
      CLOSE(5)

C Turn off window ILD
      CALL DSPOFF(2)
      RETURN
      END

C_____________________________________________________
C
C DEFINE OTHER ENTITIES
C_____________________________________________________

      SUBROUTINE GETENT
      INCLUDE SCM91.FOR

C World queue
      CALL DEFSET(IWRLDQ,'WRLD',1,3,37,0,0,4,0)

C Queue for set up change
      CALL DEFSET(ISETQ,'SETQ',0,0,0,0,0,0,0)

C Pallet entities
      DO 80 J=1,NPLTS
         CALL DEFENT(JCOMP,'    ',0)
         CALL ADLAST(JCOMP,IWRLDQ)
80    CONTINUE

C Control entities for component recording
      DO 40 N=1,NOCMP
        CALL DEFENT(JCPT(N),'CMPT',0)
        CALL ISETAT(JCPT(N),1,N)
40    CONTINUE



C General purpose entity for control details
      CALL DEFENT(JCONT,'CONT',0)

      NCOUNT=0
      IPAGE=1
```

```
      DO 10 I=1,NOCLS

C  Calculate x,y coordinates for each group
          DO 20 J=1,KNOGRP(I)

C Define dummy (transport) sets and floor waiting
queues
              NLD=KPGENM(I)*10+1
              IXVAL1=KXCORD(I,J)
              IYVAL1=KYCORD(I,J)

              CALL DEFSET(IDUMMY(I,J),'DUMM',NLD,IXVAL1-
2,
     &        IYVAL1+3,0,-1,1,0)
              CALL SETDMX(IDUMMY(I,J),4)

              CALL DEFSET(IFLOOR(I,J),'FLOR',NLD,
     &        IXVAL1+4,IYVAL1+3,-1,0,1,0)
              CALL SETDMX(IFLOOR(I,J),5)

              IF(J.EQ.KNOGRP(I)) THEN
                 NMC=KNOMCH(I,J)
                 IF(NMC.LE.4) THEN
                    N3=6
                 ELSE
                    N3=NMC+2
                 ENDIF
                 CALL DEFSET(IDUMMY(I,J+1),'DUMM',NLD,
     &           IXVAL1+N3,IYVAL1+3,0,-1,1,0)
                 CALL SETDMX(IDUMMY(I,J+1),4)
              ENDIF
              MXPGE=IPAGE

              NLD=KPGENM(I)*10+1
              DO 30 K=1,KNOMCH(I,J)
                 NCOUNT=NCOUNT+1
                 CALL DEFSET(IMACH(NCOUNT),'MACH',NLD,
     &           KXCORD(I,J)+K,KYCORD(I,J)+1,-1,0,1,1)
                 CALL DEFENT(JMACH(NCOUNT),'MACH',0)
                 CALL ISETAT(JMACH(NCOUNT),1,1)
                 CALL ISETAT(JMACH(NCOUNT),2,1)
30            CONTINUE

20        CONTINUE


10    CONTINUE

      RETURN
      END

C
C_____________________________________________________
C Zero arrays initially
C_____________________________________________________
      SUBROUTINE ZERARR
      INCLUDE SCM91.FOR
```

```
C Zero component arrays
      DO 125 I=1,MXCMP

C WIP recorders
         DO 110 J=1,4
            KCMREC(I,J)=0
110      CONTINUE

C Component state recorders
         DO 120 J=1,6
            RMCOP(I,J)=0.
120      CONTINUE

         FMXMN(I,1)=0.
C Starting minimum flow time for comparison
         FMXMN(I,2)=9999.

C Output and total flow times by route
         DO 140 J=0,MXALT+1
            CMFLW(I,J)=0.
            NCMOUT(I,J)=0
            SIGSQ(I,J)=0.0
140      CONTINUE

125    CONTINUE

C WIP at each operation
      DO 200 J=1,MOPNS
         KCMWIP(J)=0
200   CONTINUE

C Machine state recorders
      DO 300 I=1,MXMCH
         DO 285 J=1,11
            RMCOP(I,J)=0.0
285      CONTINUE
300   CONTINUE

C Batches completed on each set up and no. set up
C change recorders
      DO 310 J=1,MSETP
         KSUREC(J)=0
         KBTREC(J)=0
310   CONTINUE

      RETURN
      END

C_____________________________________________________
C
C DEFINE STAGE
C_____________________________________________________
      SUBROUTINE DEFINE
      INCLUDE SCM91.FOR

C First, read the INIT file
      CALL DINTLZ(1,1,1)
```

```
      CALL GETBAJ

      CALL SINTLZ(NOENTS,NOATTR,0,0,NOSETS,NOMEMS,
     & 0,0,0,0,0,0,ISBLKL,MAXELS)

C Zero arrays
      CALL ZERARR
C Logical displays
      CALL DFILD

C Read machine details and set up sets and entities
      CALL GETMCH

C Read component details
      CALL GETOPN

C Define entities
      CALL GETENT

C Interactions
      CALL DFOWN(1,'PRAM')
      CALL DFOWN(2,'CDSP')
      CALL DFOWN(3,'MCHQ')
      CALL DFOWN(4,'PRDQ')
      CALL DFOWN(5,'LDTM')
      CALL DFOWN(6,'REPT')

C Save file name
      CALL DFSAVE(1,'KEEP')
      RETURN
      END


C_____________________________________________________________________
C
C Define logical displays
C_____________________________________________________________________

      SUBROUTINE DFILD
      INCLUDE SCM91.FOR
C---LOGICAL DISPLAYS

C---HEADINGS
      CALL DSPCHG(1,1700)
C---UNDERLINE
      CALL DSPCHG(2,1600)
C---BOXES AND DISPLAY KEYS
      CALL DSPCHG(3,1100)
C---COMPONENTS
      CALL DSPCHG(4,1030)
      CALL DSPCHG(5,1470)
      CALL DSPCHG(6,1020)
C---OUTPUT
      CALL DSPCHG(8,2600)
      CALL DSPCHG(9,1600)
      CALL DSPCHG(10,1000)

      DO 10 I=10,90,10
```

```
      CALL DSPCHG(I+1,1000)
      CALL DSPCHG(I+2,1020)
      CALL DSPCHG(I+3,1130)
      CALL DSPCHG(I+4,1020)
      CALL DSPCHG(I+5,1160)
      CALL DSPCHG(I+6,1010)
      CALL DSPCHG(I+7,1740)
      CALL DSPCHG(I+8,1500)
      CALL DSPCHG(I+9,1350)
10    CONTINUE
      RETURN
      END
C_____________________________________________________________________
C
C Switch on logical displays
C_____________________________________________________________________
      SUBROUTINE ILDON
      INCLUDE SCM91.FOR
      DO 5 I=1,6
         CALL DISPON(I,1)
5     CONTINUE

      DO 10 I=1,9
         CALL DISPON((NOPGE*10)+I,1)
10    CONTINUE
      RETURN
      END
C_____________________________________________________________________
C
C INITIAL CONDITIONS
C_____________________________________________________________________
      SUBROUTINE INITAL
      INCLUDE SCM91.FOR

      NOPGE=1

C  Set up first breakdowns
      IJ=29001
      DO 20 I=1,NOMCH
         IF(RMTBF(I).LE.0.0) THEN
            BKDNTM=0.0
         ELSE
            DO 50 L=1,9
               IJ=IJ+4
               KSEED(I,L)=IJ
50          CONTINUE
            CALL RNDNUM(0,I,UPROB)
C MTBF is supplied in hours
            BKDNTM=-1.0*60.*RMTBF(I)*ALOG(UPROB)
         ENDIF
         RMCOP(I,1)=BKDNTM

C Set up time to first tool change
         DO 60 NSET=1,MXMCOP(I)
            IF(KMCDT1(NMCOPS(I)+NSET).EQ.0) GOTO 20
            NCOMP=KMCDT1(NMCOPS(I)+NSET)/100
            NOP=KMCDT1(NMCOPS(I)+NSET)-NCOMP*100
            TNEXT=FLOAT(KMCDT3(NMCOPS(I)+NSET))*
```

```
     &             ROPTIM(NCOMOP(NCOMP)+NOP)
              RTOLCH(NMCOPS(I)+NSET)=TNEXT
60          CONTINUE

20      CONTINUE

        RSTPD=STIME(1)

C Schedule first arrivals
        DO 30 N=1,NOCMP
C Sample and adjust for cumulative average inter
C-arrival time
          CALL SCHEDL(1,SNEGX(RARRVL(N),NRNS(N)),JCPT(N))
30      CONTINUE

        OPEN(UNIT=10,FILE='TEST.DAT')

        RETURN
        END

C
C_____________________________________________________
C Initialisation continued
C_____________________________________________________

        SUBROUTINE INITL2
        INCLUDE SCM91.FOR

C Get initial setups
        IF(SFIL.NE.'NONE') THEN
C Format: BA mc-no., Setup no,Setup batches done

        CALL BTEXT(SFIL)
        OPEN(UNIT=5,FILE=SFIL)
        DO 10 I=1,NOMCH
           READ(5,*) NBANO,ISETUP,IBATS

C Set m/c to given critical op.
           CALL ISETAT(JMACH(I),1,ISETUP)
C Set no of batches completed at the beginning
           CALL ISETAT(JMACH(I),3,IBATS)
10      CONTINUE

        ELSE

C Set m/c to first op.
        DO 40 I=1,NOMCH
           CALL ISETAT(JMACH(I),1,1)
C But allow immediate set up change if needed at the
C beginning
           CALL ISETAT(JMACH(I),3,KMCDT2(NMCOPS(I)+1))
40      CONTINUE
        ENDIF

C First despatching rule
        NRULE(1)=1
C First alternate routing rule
        NRULE(2)=1
```

```
C Replication no.
      NRULE(3)=NRNS(1)
C No. periods to warm up
      NRULE(5)=5
C No. periods in run time
      NRULE(6)=50
C Expected end of simulation time
      CALL RSETAT(JCONT,1,550001.)

C Expected period length
      CALL RSETAT(JCONT,2,10000.)
      CALL SCHEDL(9,RATTRB(JCONT,2),ISETQ)
C Schedule end of warm up period
      TWARM=FLOAT(NRULE(5))*RATTRB(JCONT,2)
      CALL SCHEDL(10,TWARM,JCONT)

C Set time update increment for screen display
      CALL SETINC(1,20.)
      CALL SETINC(2,60.)

C Set statistics for floor sets
      DO 70 I=1,NOCLS
        DO 80 J=1,KNOGRP(I)
          CALL SETSTA(IFLOOR(I,J))
80      CONTINUE
70    CONTINUE

      RETURN
      END

C_____________________________________________________
C
C SCREEN LAYOUT
C_____________________________________________________

      SUBROUTINE DSFORM
      INCLUDE SCM91.FOR

C Generalised display
      DO 50 N1=1,NOCLS
         IF(KPGENM(N1).NE.NOPGE) GOTO 50
C Cell Title
         NX=KXCORD(N1,1)
         NY=KYCORD(N1,1)
         CALL HTEXT(1,NX-1,NY+5,CNAME(N1))

         DO 60 N2=1,KNOGRP(N1)
            NX=KXCORD(N1,N2)
            NY=KYCORD(N1,N2)

C Draw box around groups
            JX1=NX*2-7
            JX2=NX*2+8
            JY1=NY*4-11
            JY2=NY*4+16
            CALL PLTLNE(3,JX1,JY1,JX1,JY2)
            CALL PLTLNE(3,JX1,JY2,JX2,JY2)
            CALL PLTLNE(3,JX1,JY1,JX2,JY1)
```

```
              CALL PLTLNE(3,JX2,JY1,JX2,JY2)

C Machine type
C              CALL HTEXT(1,NX,NY-1,CTYP(N1,N2))

60          CONTINUE
50      CONTINUE

C List interactions

        CALL HTEXT(2,3,47,'Interactions:')
        CALL HTEXT(2,3,46,'PRAM= Parameters')
        CALL HTEXT(2,3,45,'MCHQ= M/C Query    ')
        CALL HTEXT(2,3,44,'CDSP= Change Shop ')
        CALL HTEXT(2,3,43,'PRDQ= Output Query')
        CALL HTEXT(2,3,42,'LDTM= Lead times')
        CALL HTEXT(2,3,41,'REPT= Reports')


C Initialise machine display
        DO 300 I=1,NOMCH
            CALL MCNUMB(JMACH(I),MNO,IC,IG,IM)
            IF(KPGENM(IC).NE.NOPGE) GOTO 300
            IS=IATTRB(JMACH(I),1)
            IX=KXCORD(IC,IG)+IM-1
            IY=KYCORD(IC,IG)
            ISTATE=IATTRB(JMACH(I),2)
            CALL CVAREA((NOPGE*10)+ISTATE+2,IX,IX,IY,IY,
     &      STATE(IS))
300     CONTINUE
        RETURN
        END


C_______________________________________________________________________
C
C BACK TO MAIN DISPLAY
C_______________________________________________________________________

        SUBROUTINE BACKDS
        INCLUDE SCM91.FOR
        DO 120 I=1,9
          CALL DISPON(I,1)
          CALL DISPON((NOPGE*10)+I,1)
120     CONTINUE

        DO 110 I=8,10
          CALL DSPOFF(I)
110     CONTINUE


        CALL REFORM(0)
        RETURN
        END


C_______________________________________________________________________
C
```

```
C  CHANGE DISPLAY TO ANOTHER PAGE
C_____________________________________________________________
      SUBROUTINE SHPDSP
      INCLUDE SCM91.FOR
C Change main display to another page
400   CALL BTEXTI('Currently on page ',NOPGE,2)
      CALL BISUP(IRET,'Which page number?')
      IF(IRET.LT.0.OR.IRET.GT.MXPGE) GOTO 400
      IF(IRET.EQ.NOPGE) GOTO 99
      NOPGE=IRET
99    RETURN
      END
```

Appendix 1.4

Overlay 3 - Reports and Interactions

```
C_____________________________________________________
C
C  Interaction to display machine utilisation
C_____________________________________________________

      SUBROUTINE MCHQRY
      INCLUDE SCM91.FOR

C Print table of % time spent in each state by each
C machine

C Header
      TIM1=STIME(1)-RSTPD
      CALL HTEXTR(8,1,45,'Time in this period =
     & ',TIM1,9,1)
      CALL HTEXT(8,1,41,'M/C#   %IDLE   %WORK   %SETU
     & %DOWN')
      CALL HTEXT(8,40,41,'%TLCH')

C Calculate % time in each state and print on screen
      DO 10 N=1,NOMCH
         IF(N.GT.20) THEN
           CALL BISUP(IRET,'MCHQRY EXCEEDED')
           CALL INTRCT
         ELSE
           CALL HINT(8,1,41-N*2,N,4)
           DO 20 I=1,6
             IF(STIME(1).LE.0.0) THEN
                RTEMP=0.0
             ELSE
                RTEMP=100.*RMCOP(N,3+I)/(STIME(1)
     &          -RSTPD)
             ENDIF
             CALL HREAL(8,8*I,41-N*2,RTEMP,5,1)
20         CONTINUE
         ENDIF
10    CONTINUE

C Press something to return to main display
      CALL BISUP(IRET,'Press 0 to return')

      RETURN
      END

C_____________________________________________________
C
C Display component output
C_____________________________________________________

      SUBROUTINE PRDQRY
      INCLUDE SCM91.FOR

C Header
      TIM1=STIME(1)-RSTPD
      CALL HTEXTR(8,1,45,'Time in this period =
     & ',TIM1,9,1)
      CALL HTEXT(8,32,41,'------WIP------')
      CALL HTEXT(8,1,39,'COMP   FLOWT   OUTPT    ACHT')
```

```
      CALL HTEXT(8,32,39,'NOW    MIN    MAX')

C Display results
      DO 10 N=1,NOCMP
        CALL HINT(8,1,39-N*2,N,4)

C Calculate average flowtime
        IF(NCMOUT(N,0).EQ.0) THEN
          RRES=0.0
        ELSE
          RRES=CMFLW(N,0)/FLOAT(NCMOUT(N,0))
        ENDIF

        CALL HREAL(8,5,39-N*2,RRES,8,1)
C Total output so far by component
        CALL HINT(8,16,39-N*2,NCMOUT(N,0),5)

C Achievement and WIP level
        RDAY=TIM1/RATTRB(JCONT,2)

ACHT=FLOAT(KCMREC(N,2))/(RDAY*FLOAT(IATTRB(JORDR,N)))
        CALL HREAL(8,24,39-N*2,ACHT,5,2)
        CALL HINT(8,32,39-N*2,KCMREC(N,1),3)
        CALL HINT(8,37,39-N*2,KCMREC(N,5),4)
        CALL HINT(8,44,39-N*2,KCMREC(N,6),3)
10    CONTINUE

C Press something to return to main display
      CALL BISUP(IRET,'Press 1 to return')

      RETURN
      END
C_____________________________________________________
C
C Update machine state recorders
C_____________________________________________________
      SUBROUTINE UPDATM
      INCLUDE SCM91.FOR

C Update machine recorders to current states
      DO 10 N=1,NOMCH
        CALL MCNUMB(JMACH(N),MACHNO,I1,I2,I3)
        NSTATE=IATTRB(JMACH(N),2)
        WTIME=STIME(1)-RMCOP(MACHNO,3)
        RMCOP(MACHNO,3)=STIME(1)
        RMCOP(MACHNO,3+NSTATE)=RMCOP(MACHNO,3+NSTATE)
     &  +WTIME
10    CONTINUE

      RETURN
      END
C_____________________________________________________
C
C Display component leadtime breakdown
C_____________________________________________________

      SUBROUTINE LDTIME
      INCLUDE SCM91.FOR
```

```
C Headers
      TIM1=STIME(1)-RSTPD
      CALL HTEXTR(8,1,45,'Time in this period =
     & ',TIM1,9,1)

      CALL HTEXT(8,1,41,'COMP    %QUEU    %WORK    %BKDN
     & %TLCH')
      CALL HTEXT(8,40,41,'%TRAV    NOWIP    NOFIN')

C Calculate % times in each state and display
      DO 10 N=1,NOCMP
        CALL HTEXT(8,1,41-N*2,DESCR(N))

        DO 20 I=1,5
          IF(STIME(1).LE.0.0.OR.RCMOP(N,1).LE.0.0) THEN
            RTEMP=0.0
          ELSE
            RTEMP=100.*RCMOP(N,I+1)/RCMOP(N,1)
          ENDIF
          CALL HREAL(8,8*I,41-N*2,RTEMP,5,1)
20      CONTINUE

        CALL HINT(8,48,41-N*2,KCMREC(N,1),5)
        CALL HINT(8,56,41-N*2,NCMOUT(N,0),5)

10    CONTINUE

      CALL BISUP(IRET,'Hit 1 to return')

      RETURN
      END
C_____________________________________________________________________
C
C Display main parameters
C_____________________________________________________________________

      SUBROUTINE PARAM
      INCLUDE SCM91.FOR

C Text
1     CONTINUE
      CALL HTEXT(9,10,40,'Present parameters')
      CALL HTEXT(9,8,37,'1  Dispatching rule  (1-4)    =
     & ')
      CALL HTEXT(9,8,35,'2  Alternate route rule(1-6) =
     & ')
      CALL HTEXT(9,8,33,'3  Replication number         =
     & ')
      CALL HTEXT(9,8,31,'4  Travel time method         =
     & ')
      CALL HTEXT(9,8,29,'5  No. warm up periods        =
     & ')
      CALL HTEXT(9,8,27,'6  No. periods in experiment =
     & ')
      CALL HTEXT(9,8,25,'7  End of simulation          =
     & ')
```

```
      CALL HTEXT(9,8,23,'8  Period duration              =
     & ')
      CALL HTEXT(9,8,21,'9  Current travel time          =
     & ')
      CALL HTEXT(9,8,19,'10 Current set up time          =
     & ')
      CALL HTEXT(9,8,17,'11 Review interarrival
     & times?')

C Display integer values
      DO 10 N=1,6
        CALL HINT(9,39,39-N*2,NRULE(N),5)
10    CONTINUE

C Display real values
      REND=FLOAT(NRULE(5)+NRULE(6))*RATTRB(JCONT,2)
      CALL RSETAT(JCONT,1,REND)
      DO 20 N=1,2
        CALL HREAL(9,39,27-N*2,RATTRB(JCONT,N),10,1)
20    CONTINUE
      CALL HREAL(9,39,21,RTVTIM(1),10,1)
      CALL HREAL(9,39,19,RMCDT1(1),10,1)

C To alter parameters
100   CALL CLAREA(10,8,50,3,5)
      CALL ISUPL(IRET,9,8,5,'Select line (0 to
     & accept)')
110   CALL CLAREA(10,8,50,3,3)

      IF(IRET.LT.1) THEN
        GOTO 99
      ELSEIF(IRET.GT.0.AND.IRET.LE.6) THEN
        CALL ISUPL(NRET,9,8,3,'Enter new value')
        NRULE(IRET)=NRET
        CALL HINT(9,39,39-IRET*2,NRET,5)
        GOTO 90
      ELSEIF(IRET.EQ.6.OR.IRET.EQ.8) THEN
        CALL RSUPL(RRET,9,8,3,'Enter new value')
        CALL RSETAT(JCONT,IRET-6,RRET)
        CALL HREAL(9,39,39-IRET*2,RRET,10,1)
        GOTO 90
      ELSEIF(IRET.EQ.9.OR.IRET.EQ.10) THEN
        CALL RSUPL(RRET,9,8,3,'Enter new value')
        IF(IRET.EQ.9) CALL REVTRV(RRET)
        IF(IRET.EQ.10) CALL REVSET(RRET)
        CALL HREAL(9,39,21,RTVTIM(1),10,1)
        CALL HREAL(9,39,19,RMCDT1(1),10,1)
        GOTO 100
      ELSEIF(IRET.EQ.11) THEN
        CALL CLRSCR(1,0)
        CALL REVVAL
        CALL CLRSCR(1,0)
        GOTO 1
      ENDIF

C Schedule end of warming up period
90    IF(NRULE(5).GT.0) THEN
        TWARM=FLOAT(NRULE(5))*RATTRB(JCONT,2)
```

```
          CALL SCHEDL(10,TWARM,JCONT)
        ELSE
C Schedule end of run time
          REND=FLOAT(NRULE(6))*RATTRB(JCONT,2)
          CALL SCHEDL(11,REND,IWRLDQ)
        ENDIF
        CALL RSETAT(JCONT,1,REND)
        CALL HREAL(9,39,25,REND,10,1)

C Schedule end of period
        CALL SCHEDL(9,RATTRB(JCONT,2),ISETQ)

        GOTO 100

99      RETURN
        END
C_______________________________________________________
C
C  Update WIP before display or reporting
C_______________________________________________________

        SUBROUTINE UPDWIP
        INCLUDE SCM91.FOR

        DO 10 IC=1,NOCLS
          DO 20 IG=1,KNOGRP(IC)
            DO 30 N=1,3

              IF(N.EQ.1) THEN
                NUM=ISIZOF(IDUMMY(IC,IG))
                IF(NUM.EQ.0) GOTO 30
                ISET=IDUMMY(IC,IG)
              ELSEIF(N.EQ.2) THEN
                NUM=ISIZOF(IFLOOR(IC,IG))
                IF(NUM.EQ.0) GOTO 30
                ISET=IFLOOR(IC,IG)
              ELSEIF(N.EQ.3.AND.IG.EQ.KNOGRP(IC)) THEN
                NUM=ISIZOF(IDUMMY(IC,IG+1))
                IF(NUM.EQ.0) GOTO 30
                ISET=IDUMMY(IC,IG+1)
              ELSE
                GOTO 30
              ENDIF

              DO 40 IPOS=1,NUM
                JX=IDNTOF(ISET,IPOS)
                ISTATE=IATTRB(JX,5)
                CALL UPDATC(JX,ISTATE)
40            CONTINUE

30          CONTINUE
20        CONTINUE
10      CONTINUE

C Within groups, for each machine:
        DO 50 IM=1,NOMCH
           IF(ISIZOF(IMACH(IM)).EQ.0) GOTO 50
           JX=IHEDOF(IMACH(IM))
```

```
         ISTATE=IATTRB(JX,5)
         CALL UPDATC(JX,ISTATE)
50    CONTINUE

      RETURN
      END

C______________________________________________________________
C
C  End warming up period
C______________________________________________________________

      SUBROUTINE ENDWRM
      INCLUDE SCM91.FOR

C Zero off state recorders for machines
      CALL UPDATM
      DO 10 N=1,NOMCH
        DO 20 I=1,6
          RMCOP(N,I+3)=0.0
20      CONTINUE
10    CONTINUE

C Record component values and zero off
      CALL UPDWIP
      DO 30 N=1,NOCMP

C State recorders
        DO 40 I=1,5
          RCMOP(N,I+1)=0.0
40      CONTINUE
        RCMOP(N,1)=0.0

C Zero off WIP level calculators and recorders
C Add in time to now
        RCMREC(N,1)=STIME(1)
        RCMREC(N,2)=0.0

C Reset max/min WIP recorders

C Min
        KCMREC(N,5)=999
C Max
        KCMREC(N,6)=0

C Reset achievement recorders
        KCMREC(N,2)=0

        DO 60 K=1,MXALT
          NCMOUT(N,K)=0
          CMFLW(N,K)=0.0
60      CONTINUE

        NCMOUT(N,0)=0
        CMFLW(N,0)=0.0

C Max/min flowtime
        FMXMN(N,1)=0.0
```

```
          FMXMN(N,2)=9999.
30     CONTINUE

C Update set up change recorder
       DO 50 K=1,MSETP
         KBTREC(K)=0
         KSUREC(K)=0
50     CONTINUE

C Record set statistics for this period
        DO 110 I=1,NOCLS
          DO 120 J=1,KNOGRP(I)
            IS=IFLOOR(I,J)
            CALL SETSTA(IS)
120       CONTINUE
110     CONTINUE

       RSTPD=STIME(1)
       RNTM=FLOAT(NRULE(6))*RATTRB(JCONT,2)
       CALL SCHEDL(11,RNTM,JCONT)
       RETURN
       END

C_____________________________________________________
C
C  End of day or period
C_____________________________________________________

       SUBROUTINE ENDAY
       INCLUDE SCM91.FOR

C Change period counter
       NPD=NPD+1

       RETURN
       END
C_____________________________________________________
C
C  Print machine utilisation
C_____________________________________________________

       SUBROUTINE REPMCH
       INCLUDE SCM91.FOR

       DIMENSION RTEMP(7),IXX1(5),IXX2(5),NTOT1(5),
      &NTOT2(5)

C Headings and format

902    FORMAT(6X,'MCNO    %IDLE    %WORK    %SETU    %DOWN
      & %TLCH')
903    FORMAT(5X,I4,2X,5F8.1)
904    FORMAT(' ')
905    FORMAT(5X,'1. Time analysis')
906    FORMAT(5X,'2. Set up recorder')
907    FORMAT(5X,'MC  SU1  BT1  SU2  BT2  SU3  BT3  SU4
      & BT4  SU5  BT5')
```

```
908   FORMAT(5X,I2,10(I5))
911   FORMAT(8X,'B/SU',F5.1,4F10.1)

C Time analysis values
      WRITE(12,904)
      WRITE(12,905)
      WRITE(12,904)
      WRITE(12,902)
      WRITE(12,904)

      DO 10 N=1,NOMCH

        DO 20 I=1,6
          RTEMP(I)=100.*RMCOP(N,I+3)/(STIME(1)-RSTPD)
20      CONTINUE
        WRITE(12,903) N,(RTEMP(I),I=1,5)
10    CONTINUE

      DO 90 NO=1,4
        WRITE(12,904)
90    CONTINUE


C Set up recorders
      WRITE(12,906)
      WRITE(12,904)
      WRITE(12,907)
      WRITE(12,904)

      DO 30 N=1,NOMCH

C Zero off temporary arrays
        DO 110 I=1,5
          IXX1(I)=0
          IXX2(I)=0
          NTOT1(I)=0
          NTOT2(I)=0
110     CONTINUE

        II=NMCOPS(N)

          DO 70 KK=1,5
            IF(KK.GT.MXMCOP(N)) GOTO 71
            IXX1(KK)=KSUREC(II+KK)
            IXX2(KK)=KBTREC(II+KK)
            NTOT1(KK)=NTOT1(KK)+KSUREC(II+KK)
            NTOT2(KK)=NTOT2(KK)+KBTREC(II+KK)
70        CONTINUE

71        WRITE(12,908) N,(IXX1(J),IXX2(J),J=1,5)


        DO 100 J=1,5
          IF(NTOT1(J).EQ.0) THEN
            RTEMP(J)=0.
          ELSE
            RTEMP(J)=FLOAT(NTOT2(J))/FLOAT(NTOT1(J))
          ENDIF
```

```
100       CONTINUE

          WRITE(12,904)
          WRITE(12,911)  (RTEMP(J),J=1,5)

          WRITE(12,904)
30      CONTINUE

99      RETURN
        END


C_____________________________________________________________________
C
C  End of experiment
C_____________________________________________________________________

        SUBROUTINE ENDEXP
        INCLUDE SCM91.FOR

        CALL KILL

        RETURN
        END
C_____________________________________________________________________
C
C  Report main parameters
C_____________________________________________________________________
        SUBROUTINE REPPRM
        INCLUDE SCM91.FOR
        CHARACTER*30 DRULE,ARULE(10)

C Text prep
        DRULE='First come, first served '

        ARULE(1)='Ratio'
        ARULE(2)='No.jobs in partial route        '
        ARULE(3)='Workload in partial route       '
        ARULE(4)='Earliest expected finish        '
        ARULE(5)='No.jobs before bottleneck       '
        ARULE(6)='Workload before bottleneck      '
        ARULE(7)='Expected start time at b/neck   '
        ARULE(8)='Feedback on flowtime            '
        ARULE(9)='Workload + job in progress      '
        ARULE(10)='Workload + breakdown            '


C Format statements
800     FORMAT('1. Experiment details')

802     FORMAT('Dispatch rule      : ',I7,2X,A25)
803     FORMAT('Alternate route    : ',I7,2X,A25)
805     FORMAT('Travel time        : ',F7.1)
817     FORMAT('Set up time        : ',F7.1)
806     FORMAT('No. warm up periods: ',I7)
807     FORMAT('No. periods in expt: ',I7)
808     FORMAT('Period duration    : ',F7.1)
818     FORMAT('Machine data file  : ',A12)
```

```
819   FORMAT('Operation data file: ',A12)
820   FORMAT('Initialisation file: ',A12)

812   FORMAT('2. Component details')
804   FORMAT('Comp no. : ',I3,4X,'Period demand      :
     & ',I9)
809   FORMAT(18X,'Inter-arrival time: ',F9.1)
909   FORMAT(18X,'Random stream no. : ',I9)
810   FORMAT(' ')
811   FORMAT('Current simulation time: ',F9.1)

C Write statements
      WRITE(12,800)
      WRITE(12,810)
C Experiment conditions
      WRITE(12,802) NRULE(1),DRULE
      WRITE(12,803) NRULE(2),ARULE(NRULE(2))
      WRITE(12,805) RTVTIM(NCOMOP(2)+1)
      WRITE(12,817) RMCDT1(1)
      WRITE(12,806) NRULE(5)
      WRITE(12,807) NRULE(6)
      WRITE(12,808) RATTRB(JCONT,2)
      WRITE(12,818) MFIL
      WRITE(12,819) OFIL
      WRITE(12,820) DFIL
      WRITE(12,810)
      WRITE(12,811) STIME(1)
      WRITE(12,810)

C Component conditions
      DO 10 N=1,NOCMP
        WRITE(12,804) N,IATTRB(JORDR,N)
        WRITE(12,809) RARRVL(N)
        WRITE(12,909) NRNS(N)
        WRITE(12,810)
10    CONTINUE
      WRITE(12,810)
      WRITE(12,810)

      RETURN
      END


C______________________________________________________________
C
C Component production reports
C______________________________________________________________

       SUBROUTINE RPCMP1
       INCLUDE SCM91.FOR

C Write statements, headings

900    FORMAT(5X,'Component production reports ')
901    FORMAT(5X,'1. Output')
902    FORMAT(12X,'No.',10X,'By route')
903    FORMAT(5X,'Comp   Batches     1     2     3
     & Ach/t')
```

```
904     FORMAT(5X,I4,2X,I5,4X,3I6,F8.1)

999     FORMAT(' ')

        WRITE(12,900)
        WRITE(12,999)

C 1. Output summary
        WRITE(12,901)
        WRITE(12,999)
        WRITE(12,902)
        WRITE(12,903)

        DO 10 I=1,NOCMP

          ACHT=100.*FLOAT(KCMREC(I,2))/(RDAY*FLOAT
     &    (IATTRB(JORDR,I)))
          WRITE(12,904) I,NCMOUT(I,0),(NCMOUT(I,J),J=1,
     &    MXALT),ACHT

10      CONTINUE

        DO 50 N=1,3
          WRITE(12,999)
50      CONTINUE

99      RETURN
        END
C______________________________________________________
C
C  Component flowtime reports
C______________________________________________________

        SUBROUTINE RPCMP2
        INCLUDE SCM91.FOR
        DIMENSION FLOWTM(MXALT),SDV(MXCMP,0:MXALT)

910     FORMAT(5X,'2. Flowtime')
911     FORMAT(16X,'Overall',14X,'By route')
912     FORMAT(13X,18('-'),6X,17('-'))
913     FORMAT(5X,'Comp     Mean   Max    Min',6X,'1',
     &  7X,'2',7X,'3')
914     FORMAT(5X,I4,2X,3F7.0,2X,3F7.0)
915     FORMAT(15X,'Standard deviations')
918     FORMAT(13X,29('-'))
916     FORMAT(5X,'Comp    Overall     1       2
     & 3')
917     FORMAT(5X,I4,2X,4F8.1)
999     FORMAT(' ')

C 2. Flowtime recorders
        WRITE(12,910)
        WRITE(12,999)
        WRITE(12,911)
        WRITE(12,912)
        WRITE(12,913)
```

```
      DO 40 I=1,NOCMP

          DO 60 J=1,MXALT
            IF(NCMOUT(I,J).GT.0) THEN
              FLOWTM(J)=CMFLW(I,J)/FLOAT(NCMOUT(I,J))
              SADJ=FLOAT(NCMOUT(I,J)-1)
              FMEAN=FLOAT(NCMOUT(I,J))*(FLOWTM(J)**2.)
              SDV(I,J)=SQRT((SIGSQ(I,J)-FMEAN)/SADJ)
            ELSE
              FLOWTM(J)=0.0
              SDV(I,J)=0.0
            ENDIF
60        CONTINUE

C Calculate average for this period across all routes
          SADJ=FLOAT(NCMOUT(I,0)-1)
          IF(NCMOUT(I,0).GT.0) THEN
            RAVE=CMFLW(I,0)/FLOAT(NCMOUT(I,0))
            SDV(I,0)=SQRT((SIGSQ(I,0)-FLOAT(NCMOUT
     &      (I,0))
     &      *RAVE**2.)/SADJ)
          ELSE
            RAVE=0.0
            SDV(I,0)=0.0
          ENDIF

          IF(FMXMN(I,1).LE.0.) FMXMN(I,2)=0.0
          WRITE(12,914) I,RAVE,FMXMN(I,1),FMXMN(I,2),
     &    (FLOWTM(J),J=1,3)

40    CONTINUE

      WRITE(12,999)
      WRITE(12,915)
      WRITE(12,918)
      WRITE(12,916)
      DO 50 I=1,NOCMP
        WRITE(12,917) I,SDV(I,0),(SDV(I,J),J=1,3)
50    CONTINUE
      WRITE(12,999)

99    RETURN
      END
C___________________________________________________
C
C  Flowtime analysis reports
C___________________________________________________

      SUBROUTINE RPCMP3
      INCLUDE SCM91.FOR
      DIMENSION RTEMP(5)

920   FORMAT(5X,'3. Flowtime analysis')
921   FORMAT(5X,'Comp   %Queu  %Work  %WtBd  %WtTc
     & %Trav')
922   FORMAT(5X,I3,2X,5F7.1)
999   FORMAT(' ')
```

```
C 3. Flowtime analysis
      WRITE(12,920)
      WRITE(12,999)
      WRITE(12,921)
      DO 70 I=1,NOCMP

          DO 90 J=1,5
            RTEMP(J)=0.0
            IF(RCMOP(I,J+1).GT.0.) RTEMP(J)=
     &          100.*RCMOP(I,J+1)/RCMOP(I,1)
90        CONTINUE

          WRITE(12,922) I,(RTEMP(J),J=1,5)

70    CONTINUE

      WRITE(12,999)
      WRITE(12,999)

99    RETURN
      END
C_____________________________________________________
C
C  WIP reports
C_____________________________________________________

      SUBROUTINE RPCMP4
      INCLUDE SCM91.FOR

930   FORMAT(5X,'4. WIP level')
933   FORMAT(5X,'Comp   Ave   Max   Min    End')
934   FORMAT(5X,I4,F6.1,2I6,I7)
999   FORMAT(' ')

C 4. WIP level
      WRITE(12,930)
      WRITE(12,999)
      WRITE(12,933)

      DO 100 I=1,NOCMP

        IF(KCMREC(I,6).EQ.0) KCMREC(I,5)=0

C Ave WIP level throughout period
        AVEWIP=RCMREC(I,2)/(STIME(1)-RSTPD)

        WRITE(12,934) I,AVEWIP,KCMREC(I,6),
     &  KCMREC(I,5),KCMREC(I,1)

100   CONTINUE

      WRITE(12,999)
      WRITE(12,999)
99    RETURN
      END
C_____________________________________________________
C
C  Floor set statistic reporting
```

```
C_____________________________________________________________________

       SUBROUTINE RPCMP5
       INCLUDE SCM91.FOR
       DIMENSION TEMP(5)

C Print floor set statistics
940    FORMAT(5X,'5.  Floor set statistics')
941    FORMAT(6X,'Cell/Gp',3X,'Ave.sz',2X,'S.d.sz',2X,
     &'Ave.tm',2X,'NIN',3X,'NOUT')
942    FORMAT(6X,I1,3X,I1,4X,F6.1,4X,F4.1,2X,F6.1,
     & 2F7.0)
943    FORMAT(11X,I2,2X,F6.1,4X,F4.1,2X,F6.1,2F5.0)

       WRITE(12,940)
       WRITE(12,941)

       DO 10 I=1,NOCLS
         DO 20 J=1,KNOGRP(I)
           IS=IFLOOR(I,J)
           TEMP(1)=ASIZOF(IS)
           TEMP(2)=DSIZOF(IS)
           TEMP(3)=ATIMIN(IS)
           TEMP(4)=FLOAT(NINTO(IS))
           TEMP(5)=FLOAT(NOUTOF(IS))
           WRITE(12,942) I,J,(TEMP(K),K=1,5)

20       CONTINUE
10     CONTINUE

99     RETURN

       END
C_____________________________________________________________________
C
C  Revise ALL travel times
C_____________________________________________________________________

      SUBROUTINE REVTRV(RTIME)
      INCLUDE SCM91.FOR

C Change all travel times to new value
      J=NCOMOP(NOCMP)+MXCMOP(NOCMP)
      DO 10 I=1,J
        RTVTIM(I)=RTIME
10    CONTINUE

      RETURN
      END
C_____________________________________________________________________
C
C  Revise ALL seting up times
C_____________________________________________________________________

      SUBROUTINE REVSET(RTIME)
      INCLUDE SCM91.FOR

C Change all set times to new value
```

```
      J=NMCOPS(NOMCH)+MXMCOP(NOMCH)
      DO 10 I=1,J
        RMCDT1(J)=RTIME
10    CONTINUE
      RETURN
      END
C______________________________________________________________________
C
C Revise inter-arrival times
C______________________________________________________________________

      SUBROUTINE REVVAL
      INCLUDE SCM91.FOR

C Display inter-arrival times
1       CALL HTEXT(9,5,37,'Comp.no   Inter-arrival
     &  time')
        DO 10 N=1,NOCMP
          CALL HINT(9,8,37-N*2,N,2)
          CALL HREAL(9,21,37-N*2,RARRVL(N),6,1)
10      CONTINUE

C Facility to change values
2     CALL CLAREA(10,8,50,3,5)
      CALL ISUPL(NRET,9,8,5,'Select line (0 to accept)
     & ')
      IF(NRET.LT.1) GOTO 99
      IF(NRET.GT.NOCMP) GOTO 2

      CALL RSUPL(RRET,9,8,3,'Enter new value ')
      RARRVL(NRET)=RRET
      GOTO 1

99    RETURN
      END

C______________________________________________________________________
C
C Summary report control
C______________________________________________________________________

      SUBROUTINE REPSUM
      INCLUDE SCM91.FOR
      CHARACTER*10 FILNM

      RDAY=(STIME(1)-RSTPD)/RATTRB(JCONT,2)
C File control
      NF1=NRNS(1)/10
      NF2=NRNS(1)-NF1*10
      IF(NRNS(1).LE.9) THEN
        FILNM='SUM'//CHAR(NF2+48)//'.RPT'
      ELSE
        FILNM='SUM'//CHAR(NF1+48)//CHAR(NF2+48)//'.RPT'
      ENDIF

      OPEN(UNIT=12,FILE=FILNM,ACCESS='SEQUENTIAL')

      CALL REPPRM
```

```
      CALL REPMCH

      CALL RPCMP1
      CALL RPCMP2
      CALL RPCMP3
      CALL RPCMP4
      CALL RPCMP5

      CLOSE(12)

      RETURN
      END
```

Appendix 1.5

Code for generating long reports

```
C________________________________________________________________
C
C________________________________________________________________
      SUBROUTINE REPPRM
      INCLUDE SCM89.FOR
      CHARACTER*10 FILNM
      CHARACTER*25 DRULE(6),ARULE(10)

C Text prep
      DRULE(1)='First come, first served '
      DRULE(2)='Shortest imminent op     '
      DRULE(3)='Achievement ratio        '
      DRULE(4)='Urgency ratio            '
      DRULE(5)='Max remaining operations '
      DRULE(6)='Launch time              '

      DO 30 N=1,10
        ARULE(N)='RULE '//CHAR(N+48)
30    CONTINUE

C      ARULE(1)='RULE 1                   '
C      ARULE(2)='RULE 2                   '
C      ARULE(3)='RULE 3                   '
C      ARULE(4)='RULE 4                   '

C Format statements
800   FORMAT('1. Experiment details')
801   FORMAT('Experiment no.     : ',I7)
802   FORMAT('Dispatch rule      : ',I7,2X,A25)
803   FORMAT('Alternate route    : ',I7,2X,A25)
805   FORMAT('Travel time        : ',F7.1)
817   FORMAT('Set up time        : ',F7.1)
806   FORMAT('No. warm up periods: ',I7)
807   FORMAT('No. periods in expt: ',I7)
808   FORMAT('Period duration    : ',F7.1)
818   FORMAT('Machine data file  : ',A12)
819   FORMAT('Operation data file: ',A12)
820   FORMAT('Initialisation file: ',A12)

812   FORMAT('2. Component details')
804   FORMAT('Comp no. : ',I3,4X,'Period demand      :
     & ',I9)
809   FORMAT(18X,'Inter-arrival time: ',F9.1)
909   FORMAT(18X,'Random stream no. : ',I9)
810   FORMAT(' ')
811   FORMAT('Current simulation time: ',F9.1)

813   FORMAT('3. Machine summary')
814   FORMAT(6X,'Alt     B/down')
815   FORMAT('M/c   Route   Pattern')
816   FORMAT(I3,5X,I2,6X,I2)

C File details
      FILNM='PRM'//CHAR(NEXPT+48)//'.RPT'
      OPEN(UNIT=11,FILE=FILNM,ACCESS='SEQUENTIAL')

C Write statements
      WRITE(11,800)
```

```
      WRITE(11,810)
C Experiment conditions
      WRITE(11,801) NEXPT
      WRITE(11,802) NRULE(1),DRULE(NRULE(1))
      WRITE(11,803) NRULE(2),ARULE(NRULE(2))
      WRITE(11,805) RTVTIM(NCOMOP(2)+1)
      WRITE(11,817) RMCDT1(1)
      WRITE(11,806) NRULE(5)
      WRITE(11,807) NRULE(6)
      WRITE(11,808) RATTRB(JCONT,2)
      WRITE(11,818) MFIL
      WRITE(11,819) OFIL
      WRITE(11,820) DFIL
      WRITE(11,810)
      WRITE(11,810)

C Component conditions
      DO 10 N=1,NOCMP
        WRITE(11,804) N,IATTRB(JORDR,N)
        WRITE(11,809) RARRVL(N)
        WRITE(11,909) NRNS(N)
        WRITE(11,810)
10    CONTINUE
      WRITE(11,810)
      WRITE(11,810)

C Machine summary
      WRITE(11,813)
      WRITE(11,814)
      WRITE(11,815)

      DO 20 N=1,NOMCH
        WRITE(11,816) N,1,1
20    CONTINUE

      WRITE(11,810)
      WRITE(11,810)


      WRITE(11,811) STIME(1)

      CLOSE(11)

      RETURN
      END
C_____________________________________________________________________
C
C  Print machine utilisation
C_____________________________________________________________________
      SUBROUTINE REPMCH
      INCLUDE SCM89.FOR
      CHARACTER*10 FILNM
      DIMENSION RTEMP(7),IXX1(5),IXX2(5),RXX(6),
     &RXSQ(6),NTOT1(5),NTOT2(5)

      NDAY=NPD-NRULE(5)
      IF(NDAY.LT.1) GOTO 99
```

```
C Headings and format
901    FORMAT('         Machine usage report ',I2)
902    FORMAT(6X,'MCNO  PD    %IDLE    %WORK    %SETU
     & %DOWN    %BLOK    %TLCH')
903    FORMAT(5X,2I4,6F8.1)
904    FORMAT(' ')
905    FORMAT(5X,'1. Time analysis')
906    FORMAT(5X,'2. Set up recorder')
907    FORMAT(5X,'MC PD SU1 BT1 SU2 BT2 SU3 BT3 SU4 BT4
     & SU5 BT5')
908    FORMAT(5X,I2,I3,10(I4))
909    FORMAT(5X,'Mean  = ',6F8.1)
910    FORMAT(5X,'St.Dv = ',6F8.1)
911    FORMAT(3X,'Mean = ',5F8.1)

C File control
       FILNM='MUT'//CHAR(NEXPT+48)//'.RPT'
       OPEN(UNIT=12,FILE=FILNM,ACCESS='SEQUENTIAL')
       WRITE(12,901) NEXPT
       WRITE(12,904)

C Time analysis values
       WRITE(12,905)
       WRITE(12,904)
       WRITE(12,902)
       WRITE(12,904)


       DO 10 N=1,NOMCH

C Zero off each array for each machine
         DO 80 I=1,6
           RXX(I)=0.0
           RXSQ(I)=0.0
80       CONTINUE

         DO 20 M=1,NDAY
           DO 50 I=1,6
             RTEMP(I)=100.*RMCDT(N,M,I)/RATTRB(JCONT,2)
             RXX(I)=RXX(I)+RTEMP(I)
             RXSQ(I)=RXSQ(I)+RTEMP(I)**2
50         CONTINUE
           WRITE(12,903) N,M,(RTEMP(I),I=1,6)
20       CONTINUE

C Calculate mean and standard deviation
         DO 60 I=1,6
           RXX(I)=RXX(I)/NDAY
           RXSQ(I)=SQRT(RXSQ(I)/NDAY-RXX(I)**2)
60       CONTINUE

         WRITE(12,904)
         WRITE(12,909)  (RXX(I),I=1,6)
         WRITE(12,910)  (RXSQ(I),I=1,6)
         WRITE(12,904)

10     CONTINUE
```

```
      DO 90 NO=1,4
        WRITE(12,904)
90    CONTINUE


C Set up recorders
      WRITE(12,906)
      WRITE(12,904)
      WRITE(12,907)
      WRITE(12,904)

      DO 30 N=1,NOMCH

C Zero off temporary arrays
        DO 110 I=1,5
          IXX1(I)=0
          IXX2(I)=0
          NTOT1(I)=0
          NTOT2(I)=0
110     CONTINUE

        II=NMCOPS(N)
        DO 40 M=1,NDAY

          DO 70 KK=1,5
            IF(KK.GT.MXMCOP(N)) GOTO 71
            IXX1(KK)=KSULOG(M,II+KK)
            IXX2(KK)=KBTLOG(M,II+KK)
            NTOT1(KK)=NTOT1(KK)+KSULOG(M,II+KK)
            NTOT2(KK)=NTOT2(KK)+KBTLOG(M,II+KK)
70        CONTINUE

71        WRITE(12,908) N,M,(IXX1(J),IXX2(J),J=1,5)

40      CONTINUE

        DO 100 J=1,5
          IF(NTOT1(J).EQ.0) THEN
            RTEMP(J)=0.
          ELSE
            RTEMP(J)=FLOAT(NTOT2(J))/FLOAT(NTOT1(J))
          ENDIF
100     CONTINUE

        WRITE(12,904)
        WRITE(12,911) (RTEMP(J),J=1,5)

        WRITE(12,904)
30    CONTINUE


      CLOSE(12)

99    RETURN
      END
```

```
C___________________________________________________________________
C___________________________________________________________________
C___________________________________________________________________
      SUBROUTINE RPCMP1
      INCLUDE SCM89.FOR
      CHARACTER*10 FILNM
      DIMENSION NTOT(0:MXALT),RTOT(0:MXALT)


      NDAY=NPD-NRULE(5)
      IF(NDAY.LT.1) GOTO 99

C File details
      FILNM='CMP'//CHAR(NEXPT+48)//'.RPT'
      OPEN(UNIT=13,FILE=FILNM,ACCESS='SEQUENTIAL')

C Write statements, headings
      WRITE(13,900)
      WRITE(13,999)


900   FORMAT('Component production report ',I2)
901   FORMAT('1. Output')
902   FORMAT(9X,'No.',9X,'By route')
903   FORMAT('Comp Pd Batches       1    2    3
     & Ach/t')
904   FORMAT(I3,I4,3X,I3,4X,3I5,F7.1)
905   FORMAT(21X,'-----------')
906   FORMAT('Total by route',3X,3I5)
907   FORMAT('Mean =',2X,F5.1,4X,3F5.1,F7.1)
999   FORMAT(' ')

C 1. Output summary
      WRITE(13,901)
      WRITE(13,999)
      WRITE(13,902)
      WRITE(13,903)

      DO 10 I=1,NOCMP
        DO 60 J=0,MXALT
          NTOT(J)=0
60      CONTINUE
        NACH=0

        DO 20 N=1,NDAY

          DO 30 K=1,MXALT
            NTOT(K)=NTOT(K)+NFLOW(I,N,K)
            NTOT(0)=NTOT(0)+NFLOW(I,N,K)
30        CONTINUE
          NACH=NACH+KWIPR(I,N,4)
          ACHT=100.*FLOAT(KWIPR(I,N,4))/FLOAT(IATTRB
     &     (JORDR,I))

          WRITE(13,904) I,N,KWIPR(I,N,5),(NFLOW(I,N,
     &     J),J=1,MXALT),ACHT
20      CONTINUE
```

```
        WRITE(13,905)
        WRITE(13,906)  (NTOT(J),J=1,MXALT)

        DO 40 NO1=0,MXALT
          RTOT(NO1)=FLOAT(NTOT(NO1))/FLOAT(NDAY)
40      CONTINUE
        RACH=0.0
        RACH=100.*FLOAT(NACH)/(FLOAT(NDAY)*FLOAT
     &  (IATTRB(JORDR,I)))

        WRITE(13,907)  (RTOT(K),K=0,MXALT),RACH
        WRITE(13,999)
10    CONTINUE

      DO 50 N=1,3
        WRITE(13,999)
50    CONTINUE

99    RETURN
      END
C______________________________________________________________________
C
C______________________________________________________________________
      SUBROUTINE RPCMP2
      INCLUDE SCM89.FOR
      DIMENSION NTOT(0:MXALT),RTOT(0:MXALT),FLOWTM
     & (MXALT)


      NDAY=NPD-NRULE(5)
      IF(NDAY.LT.1) GOTO 99

910   FORMAT('2. Flowtime')
911   FORMAT(14X,'Overall',14X,'By route')
912   FORMAT(11X,18('-'),6X,17('-'))
913   FORMAT('Comp  Pd    Ave    Max    Min',6X,'1',
     & 7X,'2',7X,'3',4X,'Cum')
914   FORMAT(2I4,1X,3F7.0,2X,4F7.0)
915   FORMAT('Mean =',3X,F7.0,16X,3F7.0)
999   FORMAT(' ')

C 2. Flowtime recorders
      WRITE(13,910)
      WRITE(13,999)
      WRITE(13,911)
      WRITE(13,912)
      WRITE(13,913)


      DO 40 I=1,NOCMP
        NSUM=0
        RSUM=0.

        DO 130 J=1,MXALT
          NTOT(J)=0
          RTOT(J)=0.
130     CONTINUE
```

```
          DO 50 N=1,NDAY
            NTOT(0)=0
            RTOT(0)=0.

            DO 60 J=1,MXALT
              FLOWTM(J)=0.0
              IF(NFLOW(I,N,J).GT.0) FLOWTM(J)=
     &                   (I,N,J)/FLOAT(NFLOW(I,N,J))
C Accumulate for average for this period across all
C routes
              RTOT(0)=RTOT(0)+FLOWR(I,N,J)
              NTOT(0)=NTOT(0)+NFLOW(I,N,J)
C Accumumlate for average for this route through all
C periods
              RTOT(J)=RTOT(J)+FLOWR(I,N,J)
              NTOT(J)=NTOT(J)+NFLOW(I,N,J)

60          CONTINUE

C Calculate average for this period across all routes
            RAVE=0.0
            IF(NTOT(0).GT.0)  RAVE=RTOT(0)/FLOAT
     &      (NTOT(0))

            IF(RMXMN(I,N,1).LE.0.) RMXMN(I,N,2)=0.0
            WRITE(13,914) I,N,RAVE,RMXMN(I,N,1),RMXMN
     &      (I,N,2),(FLOWTM(J),J=1,3),RSCUM(I,N)
50        CONTINUE

C Calculate average for all periods for each route
          DO 120 J=1,MXALT
            FLOWTM(J)=0.0
            IF(NTOT(J).GT.0.) FLOWTM(J)=RTOT(J)/FLOAT
     &      (NTOT(J))

C Accumulate for all periods across all routes
            NSUM=NSUM+NTOT(J)
            RSUM=RSUM+RTOT(J)

120       CONTINUE
C Calculate average for all periods across all routes
          RAVE=0.0
          IF(NSUM.GT.0) RAVE=RSUM/FLOAT(NSUM)
          WRITE(13,915) RAVE,(FLOWTM(J),J=1,3)
          WRITE(13,999)

40      CONTINUE

        WRITE(13,999)

99      RETURN
        END
C____________________________________________________
C
C____________________________________________________
        SUBROUTINE RPCMP3
        INCLUDE SCM89.FOR
        DIMENSION RTEMP(5),RCUM(0:6)
```

```
      NDAY=NPD-NRULE(5)
      IF(NDAY.LT.1) GOTO 99

920   FORMAT('3. Flowtime analysis')
921   FORMAT('Comp Pd  %Queu %Work %WtBd %WtTc %Trav')
922   FORMAT(I3,I4,1X,5F6.1)
923   FORMAT('Mean =',2X,5F6.1)
999   FORMAT(' ')

C 3. Flowtime analysis
      WRITE(13,920)
      WRITE(13,999)
      WRITE(13,921)
      DO 70 I=1,NOCMP
        DO 20 K=0,6
          RCUM(K)=0.
20      CONTINUE

        DO 80 N=1,NDAY

          DO 90 J=1,5
            RTEMP(J)=0.0
            IF(RCMDT(I,N,0).GT.0.) RTEMP(J)=
     6                 100.*RCMDT(I,N,J)/RCMDT(I,N,0)
            RCUM(J)=RCUM(J)+RCMDT(I,N,J)
90        CONTINUE

          RCUM(0)=RCUM(0)+RCMDT(I,N,0)
          WRITE(13,922) I,N,(RTEMP(J),J=1,5)
80      CONTINUE

        DO 10 K=1,5
          RCUM(K)=100.*RCUM(K)/RCUM(0)
10      CONTINUE

        WRITE(13,923) (RCUM(K),K=1,5)
        WRITE(13,999)

70    CONTINUE

      WRITE(13,999)
      WRITE(13,999)

99    RETURN
      END
C_____________________________________________________________________
C
C_____________________________________________________________________
      SUBROUTINE RPCMP4
      INCLUDE SCM89.FOR

      NDAY=NPD-NRULE(5)
      IF(NDAY.LT.1) GOTO 99

930   FORMAT('4. WIP level')
933   FORMAT('Comp Pd   Ave Max Min EndPd')
```

```
934     FORMAT(I3,I4,F6.1,2I4,I5)
936     FORMAT('Mean = ',F6.1)
999     FORMAT(' ')

C 4. WIP level
        WRITE(13,930)
        WRITE(13,999)
        WRITE(13,933)

        DO 100 I=1,NOCMP
          RSUM=0.
          DO 110 N=1,NDAY

            IF(KWIPR(I,N,3).EQ.0) KWIPR(I,N,2)=0
            WRITE(13,934) I,N,RWIPR(I,N),KWIPR(I,N,3),
     &      KWIPR(I,N,2),KWIPR(I,N,1)
            RSUM=RSUM+RWIPR(I,N)
110       CONTINUE

        RSUM=RSUM/FLOAT(NDAY)
        WRITE(13,936) RSUM
        WRITE(13,999)
100     CONTINUE

        WRITE(13,999)
        WRITE(13,999)
99      RETURN
        END
C_____________________________________________________________________
C
C_____________________________________________________________________
        SUBROUTINE RPCMP5
        INCLUDE SCM89.FOR

        NDAY=NPD-NRULE(5)
        IF(NDAY.LT.1) GOTO 99

C Print floor set statistics
940     FORMAT('5.  Floor set statistics')
941     FORMAT(1X,'Cell/Gp Pd',3X,'Ave.sz',2X,'S.d.sz',
     & 2X,'Ave.tm',2X,'NIN',2X,'NOUT')
942     FORMAT(1X,I1,3X,I1,3X,I2,2X,F6.1,4X,F4.1,
     & 2X,F6.1,2F5.0)
943     FORMAT(9X,I2,2X,F6.1,4X,F4.1,2X,F6.1,2F5.0)

        WRITE(13,940)
        WRITE(13,941)

        N1=0
        DO 10 I=1,NOCLS
          DO 20 J=1,KNOGRP(I)
            N1=N1+1
            DO 5 N=1,NDAY
              IF(N.LE.1) THEN
                WRITE(13,942) I,J,N,(RSETST(N,N1,K),
     &          K=1,5)
              ELSE
                WRITE(13,943) N,(RSETST(N,N1,K),K=1,5)
```

```
             ENDIF
5          CONTINUE
20       CONTINUE
10     CONTINUE

       CLOSE(13)

99     RETURN
       END
```

Appendix 1.6

Code for single operation alternatives

```
C_____________________________________________________________
C
C Choose route from single op alternates
C_____________________________________________________________

      SUBROUTINE CHSALT(NCOMP,NO,IA3,IWHICH)
      INCLUDE SCM90.FOR
      DIMENSION IA3(MXALT+1),INM(MXALT+1),IMUCH
     &(MXALT+1)

      GOTO (100,200,300,400,500,600) NRULE(2)

C Ratio assignment strategies
C   1. Random assignment
100   SAMP=SUNIFM(0.,1.,NRNS(NOCMP)+1)
      R1=1./FLOAT(NO)
      DO 30 N=1,NO
        IF(SAMP.LE.R1*FLOAT(N)) THEN
          IWHICH=N
          GOTO 99
        ENDIF
30    CONTINUE

C    2. Ratio of pallets according to preferred route
C    or bottleneck op times
200   RMIN=10.**5.
      DO 50 N=1,NO
        NOP=IA3(N)
        INDX=NCOMOP(NCOMP)+NOP
        RTEMP=FLOAT(NRTOUT(INDX))/RATIO(INDX)
        IF(RTEMP.LT.RMIN) THEN
          RMIN=RTEMP
          IWHICH=N
        ENDIF
50    CONTINUE
      INDX=NCOMOP(NCOMP)+IA3(IWHICH)
      NRTOUT(INDX)=NRTOUT(INDX)+1
      GOTO 99

C Queue assignment strategies
C    1.   No. jobs at next stage
300   DO 20 N=1,NO
        NOP=IA3(N)
        INDX=NCOMOP(NCOMP)+NOP
        IC=KOPCEL(INDX)
        IG=KOPGRP(INDX)
        INM(N)=N
        IMUCH(N)=ISIZOF(IFLOOR(IC,IG))
20    CONTINUE
      CALL SORT(NO,INM,IMUCH)
      IWHICH=INM(1)
      GOTO 99

C    2.  Workload in queue at next stage
400   DO 70 N=1,NO
        NOP=IA3(N)
        CALL MCLOD2(NCOMP,NOP,XMUCH)
        IMUCH(N)=INT(XMUCH+0.5)
```

```
        INM(N)=N
70    CONTINUE

      CALL SORT(NO,INM,IMUCH)
      IWHICH=INM(1)
      GOTO 99

C    2.  First expected route to start work on this job
500   DO 90 N=1,NO
        NOP=IA3(N)
        CALL EARLST(NCOMP,NOP,XMUCH)
        IMUCH(N)=INT(XMUCH+0.5)
        INM(N)=N
90    CONTINUE

      CALL SORT(NO,INM,IMUCH)
      IWHICH=INM(1)
      GOTO 99

C If material has been already sent on that m/c then
C send whole batch to that route
600   DO 10 I=1,NO
        IWHICH=I
        CALL FINDMC(IA1,IA3(I),0,MNO,IST)
        IF(IST.GT.0) THEN
          ICOMP=KMCDT1(NMCOPS(MNO)+IST)
          IF(ICOMP.LE.0) GOTO 10
          ICOMP1=ICOMP/100
          ICOMP2=ICOMP-ICOMP1*100
          IF(KCMWIP(NCOMOP(ICOMP1)+ICOMP2).GT.0) GOTO
     &    99

          ENDIF
C If a m/c has been already set up for one alternate
C send whole batch to that route
          CALL FINDMC(IA1,IA3(I),1,MNO,IST)
          CALL MCNUMB(JMACH(MNO),MACHNO,IC,IG,IM)
          IF(IST.GT.0.AND.ISIZOF(IFLOOR(IC,IG)).LE.0)
     &    GOTO 99
10    CONTINUE

99    RETURN
      END


C
C______________________________________________________
C
C Find earliest expected start time for arriving job
C______________________________________________________

      SUBROUTINE EARLST(NCOMP,NOP,XMUCH2)
      INCLUDE SCM90.FOR

      XMUCH2=0.0
C MCLOD2 supplies workload on floor
      CALL MCLOD2(NCOMP,NOP,XMUCH3)
```

```
      INDX=NCOMOP(NCOMP)+NOP
      IC=KOPCEL(INDX)
      IG=KOPGRP(INDX)
      MNO=MCFIND(IC,IG,1)
      TREM=0.0
      IF(IATTRB(JMACH(MNO),2).EQ.2) THEN
C Add job currently in progress
        JX=IHEDOF(IMACH(MNO))
        TREM=TIMECL(JX)-STIME(1)
      ELSEIF(IATTRB(JMACH(MNO),2).EQ.4) THEN
C Add time to repair, if broken down plus remaining
C machining time
        TREM=TIMECL(JMACH(MNO))-STIME(1)
        TREM=TREM+RATTRB(IHEDOF(IMACH(MNO)),3)
      ENDIF

      XMUCH2=XMUCH2+XMUCH3+TREM

99    RETURN
      END
C_______________________________________________________________
C
C Assuming one machine per group, and one op alts,
C calculate workload currently waiting on the floor
C_______________________________________________________________

      SUBROUTINE MCLOD2(NCOMP,NOP,XMUCH3)
      INCLUDE SCM90.FOR

        XMUCH3=0.0
        INDX=NCOMOP(NCOMP)+NOP
C Return min. workload of machines which could do this
op
        IC=KOPCEL(INDX)
        IG=KOPGRP(INDX)
        DO 40 I=1,ISIZOF(IFLOOR(IC,IG))
          JX=IDNTOF(IFLOOR(IC,IG),I)
          NC=IATTRB(JX,1)
          NO=IATTRB(JX,2)
          XMUCH3=XMUCH3+FLOAT(KPLCAP(NC))*ROPTIM
     &    (NCOMOP(NC)+NO)
40      CONTINUE
        RETURN
        END
C_______________________________________________________________
C
C Determines current workload of a m/c
C Assumes same op. code does not appear on two
C different set-up codes of a m/c
C INPUT:MACHNO=m/c no.
C_______________________________________________________________

      SUBROUTINE MCHLOD(MNO,XMUCH1)
      INCLUDE SCM90.FOR

      XMUCH1=0.0
C Examine each op of a machine
      DO 10 N=1,MXMCOP(MNO)
```

```
          IE=KMCDT1(NMCOPS(MNO)+N)
          NCOMP=IE/100
          NOP=IE-NCOMP*100
C Wip recorder includes those in floor and those
C expected (in DUMMY)
          NJOBS=KCMWIP(NCOMOP(NCOMP)+NOP)

C Find out how many other m/c in this group could do
C this op
          CALL MCNUMB(JMACH(MNO),MACHNO,IC,IG,IM)
          NOM=0
          DO 20 M=1,KNOMCH(IC,IG)
            NMC=MCFIND(IC,IG,M)
C Can this machine do this op
            CALL FINDOP(NMC,IE,0,IST)
            IF(IST.GT.0) NOM=NOM+1
20        CONTINUE
          TSHARE=FLOAT(NJOBS)*FLOAT(KPLCAP(NCOMP))
     &    /FLOAT(NOM)
          XMUCH1=XMUCH1+TSHARE
          WRITE(10,*) 1, MNO, IE, XMUCH1
10      CONTINUE

        RETURN
        END
```

Appendix 2

Example of initialisation data file

4	Number of components
1	Number of screens in display
300	Number of pallet entities
320	Total number of entities
3	Number of attributes
25	Number of sets
10	Number of members
10	Storage block size
16300	Size of See Why array
'NONE'	Name of WIP file
'OP13F.DAT'	Name of operations file
'MC12A.DAT'	Name of machine data file
'NONE'	Name of set up file

Appendix 3

Example of machine data file

Layout of data:

'Shop name'   No. cells in shop   Starting cell no.

'Cell name'   Cell number

'Group name' Cell no., group no., No. machines in group, X coordinate, Y coordinate, identical machine marker

Mean time between failures (in hours)

For each set up code:
Code, forced minimum no. of batches, set up time, tool change quantity (not used), tool change time (not used)

-1/ indicates end of data for this group

```
'SHOP1'      6     1
'EINS'       1
'IGNORE'     1     1     1     10    25     0
 13.2

 101    3      30.    5000    5.   /
 201    2      30.    5000    5.   /
 301    1      30.    5000    5.   /
 401    1      30.    5000    5.   /
-1/

'ZWEI'       1
'IGNORE'     2     1     1     40    33     0
 13.2

 102    1      30.    5000    5.   /
 202    1      30.    5000    5.   /
 302    1      30.    5000    5.   /
 402    1      30.    5000    5.   /
-1/

'DREI'       1
'IGNORE'     3     1     1     25    17     0
 13.2

 103    1      30.    5000    5.   /
 203    1      30.    5000    5.   /
 303    1      30.    5000    5.   /
 403    1      30.    5000    5.   /
-1/

'VIER'      1
'IGNORE'    4     1     1     35    17    0
 13.2

 104    1      30.    5000    5.   /
 204    1      30.    5000    5.   /
 304    1      30.    5000    5.   /
 404    1      30.    5000    5.   /
 -1/

'FUNF'      1
'IGNORE'    5     1     1     45    17    0
 13.2

 105    1      30.    5000    5.   /
 205    1      30.    5000    5.   /
 305    1      30.    5000    5.   /
 405    1      30.    5000    5.   /
 -1/

'SECH'      1
'IGNORE'    6     1     1     60     25    0
 13.2

 106    3      30.    5000    5.    /
 206    2      30.    5000    5.    /
 306    1      30.    5000    5.    /
```

```
406   1      30.   5000   5.   /
-1/
```

Appendix 4

Example of operation data file

Layout of data:

'Component name', descriptor, comp.no., cell no., period requirement, pallet capacity, minimum WIP level (not used), maximum WIP level (not used), See WHy display no., interarrival time, arrival random stream number

For each operation:

Operation name, cell location, group location, set up no., operation time per component, travel time to next op., number of ops to skip in operation list if alternates are being used, first op on alternate branch, ratio (i.e. proportion of jobs to this alternate route), alternate op numbers

-1/ indicates end of operations for this component

```
'COMP1'   'A'   1    1   1625  50   50   2000   4   307.   2
'ONE'      1    1    1    0.63      30.    0     /
'TWO'      2    1    1    1.0       30.    3    102  0.75  103/
'ALT1'     3    1    1    0.7       30.    0    102  0.25  102/
'ALT2'     4    1    1    3.0       30.    0    102 /
'ALT3'     5    1    1    1.5       30.    0    102 /
'THREE'    6    1    1    0.63      30.    0     /
'LAST' -1/

'COMP2'   'B'   2    1   1083  50   50   2000   5   462.   3
'ONE'      1    1    2    0.94      30.    0     /
'TWO'      2    1    2    2.        30.    3    202  0.67  203/
'ALT1'     3    1    2    1.1       30.    0    202  0.33  202/
'ALT2'     4    1    2    2.3       30.    0    202 /
'ALT3'     5    1    2    4.        30.    0    202 /
'THREE'    6    1    2    0.94      30.    0     /
'LAST' -1/

'COMP3'   'C'   3    1   900   50   50   2000   6   554.   4
'ONE'      1    1    3    1.13      30.    0     /
'TWO'      2    1    3    3.6       30.    3    302  0.5  303/
'ALT1'     3    1    3    1.6       30.    0    302  0.5  302/
'ALT2'     4    1    3    3.6       30.    0    302  /
'ALT3'     5    1    3    1.0       30.    0    302  /
'THREE'    6    1    3    1.13      30.    0     /
'LAST' -1/

'COMP4'   'D'   4    1   650   50   50   2000   4   769.   5
'ONE'      1    1    4    1.56      30.    0     /
'TWO'      2    1    4    4.0       30.    3    402  0.33  403/
'ALT1'     3    1    4    0.5       30.    0    402  0.67  402/
'ALT2'     4    1    4    1.5       30.    0    402 /
'ALT3'     5    1    4    2.0       30.    0    402 /
'THREE'    6    1    4    1.56      30.    0     /
'LAST'  -1/
```

Appendix 5

Typical summary report

1. Experiment details

```
Dispatch rule       :        1  First come, first served
Alternate route     :        5  Earliest expected start
Travel time         :     30.0
Set up time         :     30.0
No. warm up periods:         5
No. periods in expt:        50
Period duration     : 10000.0
Machine data file   : MC3A.DAT
Operation data file: OP10F.DAT
Initialisation file: IN3.DAT

Current simulation time:  550000.0

Comp no. :   1     Period demand      :      1625
                   Inter-arrival time:     307.0
                   Random stream no. :         1

Comp no. :   2     Period demand      :      1083
                   Inter-arrival time:     462.0
                   Random stream no. :         2

Comp no. :   3     Period demand      :       900
                   Inter-arrival time:     554.0
                   Random stream no. :         3

Comp no. :   4     Period demand      :       650
                   Inter-arrival time:     769.0
                   Random stream no. :         4
```

1. Time analysis

MCNO	%IDLE	%WORK	%SETU	%DOWN	%TLCH
1	44.8	38.3	12.1	4.7	0.0
2	20.9	60.8	11.4	6.9	0.0
3	23.5	61.2	6.9	8.4	0.0
4	44.8	38.1	12.1	4.9	0.0

2. Set up recorder

MC	SU1	BT1	SU2	BT2	SU3	BT3	SU4	BT4	SU5	BT5
1	636	2100	533	1168	470	614	379	457	0	0
	B/SU	3.3		2.2		1.3		1.2		0.0
2	599	1098	506	706	453	574	343	375	0	0
	B/SU	1.8		1.4		1.3		1.1		0.0

3	371	588	323	419	262	330	196	249	0	0
	B/SU	1.6		1.3		1.3		1.3		0.0
4	632	2108	531	1170	482	630	371	432	0	0
	B/SU	3.3		2.2		1.3		1.2		0.0

Component production reports

1. Output

Comp	No. Batches	By route 1	2	3	Ach/t
1	1685	1097	588	0	23.0
2	1126	707	419	0	-17.1
3	904	574	330	0	-45.2
4	625	376	249	0	96.2

2. Flowtime

	Overall			By route		
Comp	Ave	Max	Min	1	2	3
1	571.	1823.	173.	530.	649.	0.
2	681.	1838.	223.	630.	765.	0.
3	740.	1873.	303.	718.	780.	0.
4	844.	2409.	353.	766.	963.	0.

3. Flowtime analysis

Comp	%Queu	%Work	%WtBd	%WtTc	%Trav
1	59.8	26.3	3.4	0.0	10.5
2	53.8	33.1	4.2	0.0	8.8
3	49.6	37.9	4.4	0.0	8.1
4	43.1	44.0	5.8	0.0	7.1

4. WIP level

Comp	Ave	Max	Min	End
1	1.9	11	0	4
2	1.5	9	0	0
3	1.3	9	0	1
4	1.1	6	0	0

5. Floor set statistics

Cell/Gp		Ave.sz	S.d.sz	Ave.tm	NIN	NOUT
1	1	1.1	1.4	131.5	4340.	4339.
2	1	0.8	1.2	153.9	2753.	2753.
3	1	0.5	0.7	148.2	1586.	1586.
4	1	0.6	0.9	73.5	4342.	4339.

Appendix 6

Typical long validation report

1. Experiment details

```
Experiment no.      :        4
Dispatch rule       :        1  First come, first served
Alternate route     :        1  RULE 1
Travel time         :     30.0
Set up time         :     30.0
No. warm up periods:         5
No. periods in expt:        25
Period duration     : 10000.0
Machine data file   : MC2A.DAT
Operation data file: OP2F.DAT
Initialisation file: IN2.DAT


Comp no. :   1     Period demand      :       1625
                   Inter-arrival time:      307.0
                   Random stream no. :          2

Comp no. :   2     Period demand      :       1083
                   Inter-arrival time:      462.0
                   Random stream no. :          3

Comp no. :   3     Period demand      :        900
                   Inter-arrival time:      554.0
                   Random stream no. :          4

Comp no. :   4     Period demand      :        650
                   Inter-arrival time:      769.0
                   Random stream no. :          5


Current simulation time: 1050000.0
```

Machine usage report 4

1. Time analysis

MCNO	PD	%IDLE	%WORK	%SETU	%DOWN	%BLOK	%TLCH
1	1	11.2	66.9	11.4	10.4	0.1	0.0
1	2	5.0	70.1	14.4	10.4	0.1	0.0
1	3	11.1	72.0	14.1	2.8	0.1	0.0
1	4	37.7	49.1	10.7	2.4	0.1	0.0
1	5	11.3	67.5	13.6	7.5	0.1	0.0
1	6	34.5	50.4	10.2	4.9	0.0	0.0
1	7	23.3	61.7	12.3	2.7	0.1	0.0
1	8	33.4	52.8	10.8	2.9	0.0	0.0
1	9	17.1	60.2	11.7	11.0	0.0	0.0
1	10	0.0	74.6	14.1	11.2	0.1	0.0
1	11	20.6	64.4	12.6	2.5	0.0	0.0
1	12	5.2	68.3	14.7	11.7	0.1	0.0
1	13	15.3	60.6	12.6	11.5	0.1	0.0
1	14	20.3	58.7	11.7	9.2	0.1	0.0
1	15	25.6	56.5	11.1	6.8	0.1	0.0
1	16	17.7	62.8	12.6	6.9	0.1	0.0
1	17	18.6	60.0	12.6	8.7	0.0	0.0
1	18	4.3	73.6	15.0	7.1	0.0	0.0

1	19	9.6	70.9	15.0	4.4	0.2	0.0
1	20	17.0	58.8	12.3	11.9	0.0	0.0
1	21	12.2	64.5	12.6	10.7	0.1	0.0
1	22	20.1	61.3	13.2	5.4	0.1	0.0
1	23	21.5	56.6	11.4	10.5	0.0	0.0
1	24	4.5	72.9	15.9	6.7	0.1	0.0
1	25	26.1	58.9	11.4	3.5	0.1	0.0
Mean	=	16.9	63.0	12.7	7.3	0.0	0.0
St.Dv	=	9.7	7.0	1.5	3.3	0.0	0.0
2	1	30.3	53.1	8.7	7.8	0.1	0.0
2	2	34.3	49.8	11.7	4.2	0.0	0.0
2	3	29.7	49.0	11.4	9.8	0.0	0.0
2	4	48.0	40.8	8.7	2.5	0.0	0.0
2	5	32.0	46.4	9.6	12.1	0.0	0.0
2	6	42.5	39.8	8.4	9.2	0.1	0.0
2	7	36.6	49.9	10.8	2.6	0.1	0.0
2	8	45.1	38.4	9.0	7.6	0.0	0.0
2	9	40.1	46.4	9.3	4.2	0.0	0.0
2	10	28.3	48.5	10.8	12.3	0.1	0.0
2	11	38.2	48.2	10.5	3.1	0.0	0.0
2	12	31.7	52.5	11.7	4.0	0.1	0.0
2	13	38.8	46.6	9.9	4.7	0.1	0.0
2	14	40.8	46.5	7.5	5.2	0.1	0.0
2	15	41.5	42.5	9.3	6.7	0.0	0.0
2	16	36.6	42.2	9.6	11.6	0.0	0.0
2	17	41.4	46.9	9.3	2.4	0.0	0.0
2	18	28.1	55.2	12.0	4.8	0.0	0.0
2	19	29.1	49.8	11.4	9.7	0.1	0.0
2	20	43.0	45.5	9.3	2.1	0.1	0.0
2	21	30.9	50.5	9.6	8.9	0.1	0.0
2	22	36.0	40.3	8.7	15.0	0.0	0.0
2	23	39.0	44.1	11.4	5.5	0.0	0.0
2	24	26.7	54.8	12.9	5.6	0.1	0.0
2	25	45.6	41.8	9.6	3.1	0.0	0.0
Mean	=	36.6	46.8	10.0	6.6	0.0	0.0
St.Dv	=	6.1	4.6	1.3	3.6	0.0	0.0
3	1	51.2	33.0	7.0	8.9	0.0	0.0
3	2	44.7	40.1	9.0	6.1	0.1	0.0
3	3	45.0	41.3	9.3	4.3	0.1	0.0
3	4	65.0	24.7	6.3	3.9	0.1	0.0
3	5	46.3	38.9	7.2	7.6	0.0	0.0
3	6	62.3	30.1	4.8	2.7	0.0	0.0
3	7	50.3	33.4	6.3	10.0	0.0	0.0
3	8	61.1	30.7	4.8	3.4	0.0	0.0
3	9	57.2	31.8	6.0	4.9	0.1	0.0
3	10	40.0	45.8	10.8	3.3	0.0	0.0
3	11	54.8	36.0	9.0	0.3	0.0	0.0
3	12	48.3	36.5	8.1	7.0	0.0	0.0
3	13	55.4	33.0	7.8	3.8	0.1	0.0
3	14	57.9	27.8	6.6	7.6	0.1	0.0
3	15	59.4	31.9	6.9	1.8	0.1	0.0
3	16	49.3	37.5	8.4	4.7	0.1	0.0
3	17	56.8	31.7	7.5	4.0	0.0	0.0
3	18	48.0	37.0	8.7	6.3	0.0	0.0
3	19	48.5	38.9	7.8	4.8	0.0	0.0

3	20	57.9	30.4	6.9	4.7	0.0	0.0
3	21	47.3	36.1	7.2	9.4	0.0	0.0
3	22	48.1	38.8	8.4	4.6	0.1	0.0
3	23	62.8	30.2	6.0	1.0	0.0	0.0
3	24	52.6	35.9	8.7	2.8	0.0	0.0
3	25	55.2	33.9	7.2	3.7	0.0	0.0
Mean	=	53.0	34.6	7.5	4.9	0.0	0.0
St.Dv	=	6.4	4.6	1.4	2.4	0.0	0.0
4	1	12.1	66.5	12.3	9.1	0.0	0.0
4	2	5.1	69.6	13.4	11.8	0.1	0.0
4	3	6.9	71.0	15.4	6.6	0.0	0.0
4	4	26.5	50.0	10.2	13.2	0.1	0.0
4	5	4.9	68.1	14.1	12.9	0.1	0.0
4	6	33.8	52.4	10.8	3.0	0.0	0.0
4	7	15.8	62.1	12.9	9.2	0.0	0.0
4	8	23.5	56.1	12.3	8.0	0.1	0.0
4	9	24.2	59.8	11.7	4.2	0.1	0.0
4	10	2.4	73.7	14.7	9.2	0.0	0.0
4	11	8.9	66.5	12.0	12.5	0.1	0.0
4	12	11.9	67.6	13.8	6.7	0.0	0.0
4	13	21.4	61.4	12.6	4.5	0.1	0.0
4	14	29.1	54.8	11.1	4.9	0.1	0.0
4	15	25.5	60.1	11.1	3.3	0.0	0.0
4	16	23.1	61.0	11.9	4.0	0.0	0.0
4	17	14.0	60.3	13.0	12.7	0.1	0.0
4	18	2.1	70.7	15.6	11.5	0.1	0.0
4	19	6.1	70.5	13.5	9.9	0.1	0.0
4	20	19.9	60.7	12.6	6.7	0.1	0.0
4	21	16.1	62.3	12.8	8.8	0.0	0.0
4	22	13.0	63.1	13.9	9.9	0.1	0.0
4	23	23.6	55.9	11.7	8.8	0.1	0.0
4	24	5.9	68.0	15.3	10.8	0.0	0.0
4	25	19.4	62.2	11.7	6.7	0.1	0.0
Mean	=	15.8	63.0	12.8	8.4	0.0	0.0
St.Dv	=	8.9	6.1	1.4	3.1	0.0	0.0

2. Set up recorder

MC	PD	SU1	BT1	SU2	BT2	SU3	BT3	SU4	BT4	SU5	BT5
1	1	11	34	9	18	6	11	12	21	0	0
1	2	12	37	11	20	14	20	11	15	0	0
1	3	10	29	13	28	14	20	10	15	0	0
1	4	11	35	6	12	11	13	7	9	0	0
1	5	9	27	10	24	15	23	12	12	0	0
1	6	12	38	7	15	10	13	5	7	0	0
1	7	14	44	8	16	12	21	7	7	0	0
1	8	11	38	10	19	10	13	5	6	0	0
1	9	9	29	9	23	12	18	9	10	0	0
1	10	10	30	13	26	14	22	10	16	0	0
1	11	12	38	10	22	14	21	6	8	0	0

1	12	13	39	12	24	13	18	11	12	0	0
1	13	11	32	9	20	14	19	8	10	0	0
1	14	12	34	8	19	9	12	10	13	0	0
1	15	9	32	10	24	9	13	9	9	0	0
1	16	9	27	12	27	13	17	8	10	0	0
1	17	11	34	8	17	11	12	12	16	0	0
1	18	13	37	10	20	15	19	12	19	0	0
1	19	10	34	12	26	15	19	13	13	0	0
1	20	9	28	9	19	14	18	9	12	0	0
1	21	14	44	11	23	7	9	10	14	0	0
1	22	11	31	9	22	16	20	8	9	0	0
1	23	10	35	10	25	8	9	10	9	0	0
1	24	12	35	15	31	16	20	10	12	0	0
1	25	12	37	11	24	9	15	6	7	0	0
Mean =			3.1		2.2		1.4		1.3		0.0
2	1	8	19	6	7	6	10	9	15	0	0
2	2	10	14	10	11	11	15	8	8	0	0
2	3	11	13	7	8	9	10	11	15	0	0
2	4	7	16	9	13	7	7	6	7	0	0
2	5	7	13	9	13	8	10	8	9	0	0
2	6	8	12	7	9	6	8	7	10	0	0
2	7	10	15	5	5	12	18	9	10	0	0
2	8	11	21	4	5	10	12	5	4	0	0
2	9	9	16	6	7	8	11	8	11	0	0
2	10	9	16	11	11	8	12	8	10	0	0
2	11	10	18	9	10	8	13	8	8	0	0
2	12	11	22	10	10	8	10	10	12	0	0
2	13	8	12	8	13	8	10	9	10	0	0
2	14	7	17	6	8	6	9	6	11	0	0
2	15	8	11	10	14	5	7	8	10	0	0
2	16	8	10	8	13	8	8	8	9	0	0
2	17	6	12	10	15	8	12	7	8	0	0
2	18	12	21	7	10	11	11	10	13	0	0
2	19	8	13	9	14	11	11	10	10	0	0
2	20	8	15	7	11	9	9	7	10	0	0
2	21	11	20	7	6	6	8	8	15	0	0
2	22	8	13	8	11	9	13	4	5	0	0
2	23	12	17	11	14	7	7	8	8	0	0
2	24	10	14	12	15	12	14	9	10	0	0
2	25	8	15	9	11	9	11	6	6	0	0
Mean =			1.7		1.3		1.3		1.2		0.0
3	1	8	12	7	10	4	6	5	5	0	0
3	2	10	20	8	9	8	7	4	8	0	0
3	3	8	15	11	17	9	11	3	2	0	0
3	4	7	12	6	7	6	6	2	3	0	0
3	5	7	12	6	10	8	14	3	4	0	0
3	6	5	14	4	9	6	10	1	1	0	0
3	7	8	25	7	9	5	7	1	1	0	0
3	8	4	11	4	12	5	7	3	3	0	0
3	9	6	9	7	13	6	11	1	1	0	0
3	10	8	12	12	16	9	11	7	7	0	0
3	11	11	15	8	9	9	13	2	2	0	0
3	12	9	15	9	13	6	6	3	5	0	0
3	13	8	16	9	10	7	10	2	2	0	0

3	14	8	14	5	8	6	6	3	3	0	0
3	15	8	16	7	8	6	10	2	2	0	0
3	16	8	16	8	13	8	9	4	3	0	0
3	17	10	15	7	8	4	4	4	7	0	0
3	18	9	16	9	12	7	7	4	5	0	0
3	19	8	18	8	12	7	9	3	4	0	0
3	20	8	10	6	9	6	7	3	5	0	0
3	21	11	18	8	16	3	6	2	2	0	0
3	22	9	15	7	9	8	10	4	6	0	0
3	23	7	11	7	13	4	6	2	3	0	0
3	24	10	20	10	13	7	7	2	2	0	0
3	25	8	20	9	13	5	7	1	1	0	0
Mean =			1.9		1.5		1.3		1.2		0.0
4	1	10	31	10	20	7	13	14	19	0	0
4	2	13	40	8	16	12	19	11	17	0	0
4	3	11	33	12	25	15	20	14	14	0	0
4	4	11	34	8	17	8	10	7	9	0	0
4	5	9	25	10	24	16	24	12	13	0	0
4	6	9	31	9	17	11	14	7	9	0	0
4	7	14	44	7	17	14	19	8	8	0	0
4	8	13	40	9	18	12	16	7	7	0	0
4	9	9	30	10	21	11	17	9	11	0	0
4	10	9	28	12	24	16	22	12	17	0	0
4	11	13	38	10	22	11	24	6	8	0	0
4	12	13	39	11	24	10	14	12	14	0	0
4	13	10	31	10	21	13	19	9	10	0	0
4	14	11	35	8	20	7	9	11	11	0	0
4	15	10	30	10	26	9	14	8	11	0	0
4	16	9	27	12	27	12	18	6	9	0	0
4	17	10	30	11	22	12	14	11	13	0	0
4	18	13	40	10	20	17	20	12	14	0	0
4	19	10	32	12	28	13	17	10	14	0	0
4	20	10	28	10	21	11	17	11	13	0	0
4	21	13	42	11	23	9	9	9	13	0	0
4	22	11	33	12	24	18	23	6	6	0	0
4	23	11	35	11	26	9	9	8	9	0	0
4	24	10	31	15	31	15	18	11	10	0	0
4	25	13	41	10	22	11	17	5	8	0	0
Mean =			3.1		2.2		1.4		1.2		0.0

Component production report

1. Output

Comp	Pd	No. Batches	By route 1	2	3	Ach/t
1	1	29	29	0	0	89.2
1	2	36	36	0	0	110.8
1	3	28	28	0	0	86.2
1	4	29	29	0	0	89.2
1	5	25	25	0	0	76.9
1	6	24	24	0	0	73.8
1	7	40	40	0	0	123.1
1	8	35	35	0	0	107.7
1	9	26	26	0	0	80.0
1	10	28	28	0	0	86.2
1	11	33	33	0	0	101.5
1	12	38	38	0	0	116.9
1	13	27	27	0	0	83.1
1	14	29	29	0	0	89.2
1	15	28	28	0	0	86.2
1	16	25	25	0	0	76.9
1	17	23	23	0	0	70.8
1	18	40	40	0	0	123.1
1	19	29	29	0	0	89.2
1	20	25	25	0	0	76.9
1	21	37	37	0	0	113.8
1	22	30	30	0	0	92.3
1	23	28	28	0	0	86.2
1	24	31	31	0	0	95.4
1	25	36	36	0	0	110.8
Total by route			759	0	0	
Mean =		30.4	30.4	0.0	0.0	
2	1	18	18	0	0	83.1
2	2	17	17	0	0	78.5
2	3	27	27	0	0	124.7
2	4	19	19	0	0	87.7
2	5	23	23	0	0	106.2
2	6	18	18	0	0	83.1
2	7	13	13	0	0	60.0
2	8	17	17	0	0	78.5
2	9	22	22	0	0	101.6
2	10	24	24	0	0	110.8
2	11	23	23	0	0	106.2
2	12	22	22	0	0	101.6
2	13	20	20	0	0	92.3
2	14	17	17	0	0	78.5
2	15	25	25	0	0	115.4
2	16	25	25	0	0	115.4
2	17	23	23	0	0	106.2
2	18	19	19	0	0	87.7
2	19	26	26	0	0	120.0
2	20	20	20	0	0	92.3
2	21	22	22	0	0	101.6
2	22	21	21	0	0	97.0
2	23	25	25	0	0	115.4

2	24	28	28	0	0	129.3
2	25	24	24	0	0	110.8
			-----	-----	-----	
Total by route			538	0	0	
Mean =		21.5	21.5	0.0	0.0	99.4
3	1	15	15	0	0	83.3
3	2	21	21	0	0	116.7
3	3	21	21	0	0	116.7
3	4	11	11	0	0	61.1
3	5	24	24	0	0	133.3
3	6	17	17	0	0	94.4
3	7	24	24	0	0	133.3
3	8	20	20	0	0	111.1
3	9	19	19	0	0	105.6
3	10	22	22	0	0	122.2
3	11	26	26	0	0	144.4
3	12	16	16	0	0	88.9
3	13	21	21	0	0	116.7
3	14	14	14	0	0	77.8
3	15	16	16	0	0	88.9
3	16	19	19	0	0	105.6
3	17	16	16	0	0	88.9
3	18	20	20	0	0	111.1
3	19	19	19	0	0	105.6
3	20	19	19	0	0	105.6
3	21	12	12	0	0	66.7
3	22	24	24	0	0	133.3
3	23	14	14	0	0	77.8
3	24	20	20	0	0	111.1
3	25	20	20	0	0	111.1
			-----	-----	-----	
Total by route			470	0	0	
Mean =		18.8	18.8	0.0	0.0	104.4
4	1	21	21	0	0	161.5
4	2	18	18	0	0	138.5
4	3	16	16	0	0	123.1
4	4	11	11	0	0	84.6
4	5	14	14	0	0	107.7
4	6	12	12	0	0	92.3
4	7	11	11	0	0	84.6
4	8	9	9	0	0	69.2
4	9	12	12	0	0	92.3
4	10	17	17	0	0	130.8
4	11	10	10	0	0	76.9
4	12	15	15	0	0	115.4
4	13	13	13	0	0	100.0
4	14	15	15	0	0	115.4
4	15	12	12	0	0	92.3
4	16	12	12	0	0	92.3
4	17	17	17	0	0	130.8
4	18	15	15	0	0	115.4
4	19	17	17	0	0	130.8
4	20	15	15	0	0	115.4
4	21	16	16	0	0	123.1
4	22	11	11	0	0	84.6
4	23	12	12	0	0	92.3

4	24	11	11	0	0	84.6
4	25	8	8	0	0	61.5
Total by route			340	0	0	
Mean =		13.6	13.6	0.0	0.0	104.6

2. Flowtime

		Overall			By route			
Comp	Pd	Ave	Max	Min	1	2	3	Cum
1	1	777.	1307.	223.	777.	0.	0.	777.
1	2	1204.	1829.	472.	1204.	0.	0.	1013.
1	3	1050.	1889.	248.	1050.	0.	0.	1024.
1	4	788.	1389.	223.	788.	0.	0.	968.
1	5	1170.	1664.	305.	1170.	0.	0.	1002.
1	6	515.	1341.	223.	515.	0.	0.	934.
1	7	790.	1563.	279.	790.	0.	0.	907.
1	8	732.	1578.	223.	732.	0.	0.	882.
1	9	739.	1442.	223.	739.	0.	0.	868.
1	10	2485.	3396.	1539.	2485.	0.	0.	1019.
1	11	1796.	3131.	278.	1796.	0.	0.	1096.
1	12	1012.	1613.	266.	1012.	0.	0.	1088.
1	13	776.	1360.	253.	776.	0.	0.	1066.
1	14	961.	1666.	223.	961.	0.	0.	1059.
1	15	695.	1152.	223.	695.	0.	0.	1037.
1	16	852.	2164.	248.	852.	0.	0.	1027.
1	17	943.	1548.	278.	943.	0.	0.	1023.
1	18	1586.	2262.	887.	1586.	0.	0.	1065.
1	19	1432.	2391.	223.	1432.	0.	0.	1083.
1	20	824.	1437.	223.	824.	0.	0.	1073.
1	21	1200.	1969.	303.	1200.	0.	0.	1080.
1	22	1072.	1676.	318.	1072.	0.	0.	1080.
1	23	826.	1781.	223.	826.	0.	0.	1069.
1	24	1093.	2024.	376.	1093.	0.	0.	1070.
1	25	685.	1890.	223.	685.	0.	0.	1052.
Mean =		1052.			1052.	0.	0.	
2	1	910.	1439.	490.	910.	0.	0.	910.
2	2	1406.	1923.	459.	1406.	0.	0.	1151.
2	3	1129.	1774.	309.	1129.	0.	0.	1141.
2	4	809.	1454.	314.	809.	0.	0.	1063.
2	5	1331.	1669.	792.	1331.	0.	0.	1123.
2	6	585.	1420.	284.	585.	0.	0.	1043.
2	7	1001.	1832.	393.	1001.	0.	0.	1039.
2	8	832.	1520.	284.	832.	0.	0.	1016.
2	9	713.	1359.	284.	713.	0.	0.	978.
2	10	2603.	3231.	1645.	2603.	0.	0.	1175.
2	11	2137.	3028.	816.	2137.	0.	0.	1275.
2	12	1338.	1854.	875.	1338.	0.	0.	1281.
2	13	1219.	1644.	284.	1219.	0.	0.	1276.
2	14	1392.	2107.	344.	1392.	0.	0.	1283.
2	15	798.	1345.	223.	798.	0.	0.	1243.
2	16	884.	1461.	304.	884.	0.	0.	1216.
2	17	873.	1757.	279.	873.	0.	0.	1194.

2	18	1800.	2251.	958.	1800.	0.	0.	1225.
2	19	1678.	2604.	339.	1678.	0.	0.	1254.
2	20	821.	1265.	284.	821.	0.	0.	1234.
2	21	1270.	1962.	586.	1270.	0.	0.	1235.
2	22	1109.	1920.	309.	1109.	0.	0.	1230.
2	23	1380.	2086.	578.	1380.	0.	0.	1237.
2	24	1336.	1842.	846.	1336.	0.	0.	1243.
2	25	864.	1867.	284.	864.	0.	0.	1226.
Mean =		1226.			1226.	0.	0.	
3	1	886.	1280.	303.	886.	0.	0.	886.
3	2	1585.	2313.	948.	1585.	0.	0.	1293.
3	3	1580.	1965.	1112.	1580.	0.	0.	1399.
3	4	1218.	1607.	502.	1218.	0.	0.	1370.
3	5	1526.	1972.	1058.	1526.	0.	0.	1410.
3	6	1067.	2348.	295.	1067.	0.	0.	1357.
3	7	1121.	2072.	539.	1121.	0.	0.	1314.
3	8	1114.	1811.	303.	1114.	0.	0.	1288.
3	9	977.	1417.	303.	977.	0.	0.	1254.
3	10	2860.	3481.	1574.	2860.	0.	0.	1436.
3	11	2024.	2940.	569.	2024.	0.	0.	1506.
3	12	1343.	1736.	465.	1343.	0.	0.	1494.
3	13	1333.	2095.	273.	1333.	0.	0.	1481.
3	14	1155.	1803.	303.	1155.	0.	0.	1464.
3	15	885.	1645.	353.	885.	0.	0.	1432.
3	16	1150.	2092.	353.	1150.	0.	0.	1415.
3	17	1179.	2066.	303.	1179.	0.	0.	1403.
3	18	1930.	2603.	1006.	1930.	0.	0.	1434.
3	19	1630.	2490.	908.	1630.	0.	0.	1444.
3	20	1183.	1808.	615.	1183.	0.	0.	1431.
3	21	1060.	1781.	273	1060.	0.	0.	1420.
3	22	1440.	2458.	593	1440.	0.	0.	1421.
3	23	972.	2009.	303	972.	0.	0.	1406.
3	24	1385.	2039.	653	1385.	0.	0.	1405.
3	25	989.	1842.	273	989.	0.	0.	1388.
Mean =		1388.			1388.	0.	0.	
4	1	1122.	1823.	326	1122.	0.	0.	1122.
4	2	1602.	2353.	1102	1602.	0.	0.	1343.
4	3	1617.	2194.	724	1617.	0.	0.	1423.
4	4	988.	1408.	466	988.	0.	0.	1350.
4	5	1439.	1794.	631	1439.	0.	0.	1366.
4	6	840.	1568.	346	840.	0.	0.	1297.
4	7	1026.	1640.	371	1026.	0.	0.	1268.
4	8	1297.	2334.	653	1297.	0.	0.	1271.
4	9	1032.	1542.	346	1032.	0.	0.	1248.
4	10	2745.	3324.	1863	2745.	0.	0.	1428.
4	11	1725.	3162.	346	1725.	0.	0.	1448.
4	12	1355.	2051.	451	1355.	0.	0.	1439.
4	13	1182.	2034.	346	1182.	0.	0.	1421.
4	14	1390.	2369.	316	1390.	0.	0.	1418.
4	15	1061.	1372.	519	1061.	0.	0.	1398.
4	16	1164.	1939.	419	1164.	0.	0.	1385.
4	17	1401.	2084.	897	1401.	0.	0.	1386.
4	18	1956.	2590.	1246	1956.	0.	0.	1420.
4	19	1879.	2386.	346	1879.	0.	0.	1449.
4	20	1069.	1885.	346	1069.	0.	0.	1429.
4	21	1333.	1928.	426	1333.	0.	0.	1424.

4	22	1213.	2146.	341	1213.	0.	0.	1416.
4	23	1142.	2157.	556	1142.	0.	0.	1406.
4	24	1463.	1720.	1148.	1463.	0.	0.	1408.
4	25	1217.	2047.	904.	1217.	0.	0.	1404.
Mean =		1404.			1404.	0.	0.	

3. Flowtime analysis

Comp	Pd	%Queu	%Work	%WtBd	%WtTc	%Trav
1	1	65.9	21.5	4.7	0.0	7.9
1	2	80.0	13.3	1.9	0.0	4.8
1	3	77.7	15.8	0.7	0.0	5.7
1	4	68.3	21.3	2.8	0.0	7.7
1	5	77.9	14.0	2.9	0.0	5.1
1	6	54.1	32.3	1.8	0.0	11.8
1	7	71.2	20.3	1.2	0.0	7.4
1	8	66.0	23.6	2.1	0.0	8.2
1	9	68.0	20.0	4.9	0.0	7.1
1	10	90.8	5.9	1.1	0.0	2.2
1	11	82.6	11.9	1.1	0.0	4.3
1	12	76.7	16.3	1.1	0.0	5.9
1	13	67.8	21.4	2.9	0.1	7.8
1	14	74.4	17.1	2.1	0.0	6.3
1	15	65.1	23.1	3.5	0.0	8.3
1	16	69.9	20.0	2.8	0.0	7.3
1	17	79.7	13.2	2.2	0.0	4.9
1	18	82.7	11.3	1.9	0.0	4.1
1	19	83.3	11.6	0.7	0.0	4.3
1	20	67.8	20.9	3.8	0.1	7.5
1	21	77.3	13.3	4.4	0.0	5.0
1	22	75.0	15.7	3.7	0.0	5.6
1	23	69.7	19.8	3.2	0.0	7.3
1	24	78.4	14.6	1.6	0.0	5.5
1	25	66.4	23.1	2.3	0.0	8.2
Mean =		76.4	15.6	2.2	0.0	5.7
2	1	62.5	26.9	4.0	0.0	6.5
2	2	74.9	17.6	3.1	0.0	4.5
2	3	71.4	21.5	1.8	0.0	5.3
2	4	62.8	25.2	5.3	0.1	6.6
2	5	72.4	19.3	3.5	0.0	4.8
2	6	43.8	40.0	5.6	0.0	10.6
2	7	68.2	22.6	3.3	0.0	5.8
2	8	61.0	30.0	1.5	0.0	7.5
2	9	64.7	27.3	1.0	0.0	6.9
2	10	87.6	8.9	1.3	0.0	2.2
2	11	82.4	13.5	0.9	0.0	3.3
2	12	78.4	16.1	1.5	0.0	3.9
2	13	70.1	22.2	2.0	0.0	5.7
2	14	75.2	18.1	2.4	0.0	4.4
2	15	63.4	28.3	1.1	0.0	7.1
2	16	57.3	28.3	7.3	0.0	7.1
2	17	64.0	26.0	3.3	0.0	6.7
2	18	84.8	11.5	0.8	0.0	2.8
2	19	74.6	17.1	4.1	0.0	4.2
2	20	60.0	29.6	2.9	0.0	7.5
2	21	73.7	18.3	3.4	0.0	4.5

2	22	69.1	22.3	2.9	0.1	5.6
2	23	75.6	16.7	3.3	0.0	4.3
2	24	74.6	18.9	1.9	0.0	4.6
2	25	61.3	28.3	3.3	0.0	7.1
Mean =		73.0	19.6	2.6	0.0	4.9
3	1	58.0	30.6	5.0	0.0	6.4
3	2	76.6	18.0	1.5	0.0	3.8
3	3	76.5	18.2	1.5	0.0	3.7
3	4	70.4	23.5	1.1	0.0	4.9
3	5	72.9	18.9	4.2	0.0	4.0
3	6	62.7	30.1	1.0	0.0	6.2
3	7	67.0	24.2	3.6	0.0	5.2
3	8	61.6	25.7	7.2	0.0	5.4
3	9	63.7	27.6	3.0	0.0	5.7
3	10	87.9	9.0	1.3	0.0	1.9
3	11	78.0	17.4	1.0	0.0	3.6
3	12	71.6	19.1	5.2	0.0	4.1
3	13	68.8	23.3	3.0	0.0	4.9
3	14	67.0	24.5	3.0	0.0	5.4
3	15	61.8	28.2	4.1	0.0	5.9
3	16	64.7	27.9	1.5	0.0	5.8
3	17	69.9	21.9	3.7	0.0	4.5
3	18	80.9	14.4	1.7	0.0	3.0
3	19	75.6	19.6	0.4	0.0	4.3
3	20	66.7	25.2	2.9	0.0	5.1
3	21	75.1	20.2	0.2	0.0	4.5
3	22	67.9	22.1	5.5	0.0	4.5
3	23	68.9	23.9	2.0	0.0	5.2
3	24	69.7	21.7	4.0	0.0	4.7
3	25	63.7	27.9	2.4	0.1	5.9
Mean =		72.5	20.5	2.6	0.0	4.3
4	1	55.4	34.4	4.9	0.0	5.2
4	2	68.0	24.0	4.1	0.0	3.8
4	3	66.1	25.4	4.5	0.0	4.0
4	4	57.3	33.3	4.4	0.0	5.0
4	5	62.9	29.6	3.1	0.0	4.3
4	6	37.9	44.0	10.6	0.0	7.5
4	7	56.6	33.1	4.6	0.0	5.7
4	8	64.7	29.9	0.6	0.0	4.8
4	9	64.5	27.9	3.4	0.0	4.3
4	10	80.7	15.1	1.9	0.0	2.3
4	11	66.7	25.2	3.9	0.0	4.1
4	12	63.5	28.7	3.3	0.0	4.4
4	13	62.0	29.2	3.9	0.0	4.8
4	14	62.9	27.6	5.0	0.1	4.4
4	15	54.0	37.4	2.9	0.0	5.7
4	16	60.1	32.8	2.0	0.0	5.1
4	17	67.3	25.7	2.9	0.0	4.1
4	18	75.4	19.1	2.5	0.0	3.1
4	19	68.7	24.1	3.7	0.0	3.5
4	20	52.0	37.2	5.1	0.0	5.8
4	21	66.4	26.5	2.8	0.0	4.2
4	22	60.2	33.4	0.8	0.0	5.6
4	23	63.1	30.3	2.1	0.0	4.5
4	24	65.2	29.0	1.2	0.0	4.5
4	25	57.1	37.5	0.0	0.0	5.4

Mean = 65.0 27.4 3.3 0.0 4.3

4. Floor set statistics

Cell/Gp		Pd	Ave.sz	S.d.sz	Ave.tm	NIN	NOUT
1	1	1	3.0	1.9	336.2	90.	84.
		2	5.4	2.2	566.3	96.	92.
		3	5.3	1.7	553.7	96.	92.
		4	1.6	1.8	211.4	76.	68.
		5	3.3	2.0	366.9	91.	87.
		6	1.3	1.3	177.1	72.	72.
		7	2.2	1.3	251.2	88.	88.
		8	1.7	1.6	220.5	77.	77.
		9	4.1	3.4	431.7	95.	80.
		10	20.2	2.2	1754.8	115.	94.
		11	6.6	6.9	737.3	89.	89.
		12	7.4	3.4	728.7	101.	93.
		13	4.4	2.0	508.2	86.	80.
		14	4.8	2.6	598.8	81.	79.
		15	3.0	2.2	352.6	84.	77.
		16	3.1	2.4	373.3	84.	82.
		17	4.5	3.9	508.7	89.	79.
		18	8.1	2.3	794.8	102.	94.
		19	7.1	3.8	754.6	94.	93.
		20	3.0	2.5	394.5	77.	77.
		21	4.9	2.8	534.2	91.	89.
		22	3.2	2.2	390.9	83.	83.
		23	4.7	4.6	572.2	83.	78.
		24	6.2	2.6	628.7	99.	98.
		25	2.7	2.3	316.1	87.	83.
2	1	1	0.2	0.5	28.4	85.	85.
		2	0.3	0.4	27.5	92.	92.
		3	0.2	0.4	26.9	92.	91.
		4	0.2	0.4	28.8	69.	69.
		5	0.3	0.5	34.9	87.	87.
		6	0.2	0.4	22.8	72.	72.
		7	0.2	0.4	24.7	89.	89.
		8	0.2	0.4	23.1	76.	76.
		9	0.2	0.4	20.1	79.	79.
		10	0.3	0.5	32.4	95.	95.
		11	0.2	0.4	21.8	89.	89.
		12	0.3	0.5	28.6	92.	92.
		13	0.2	0.4	24.7	82.	82.
		14	0.2	0.4	21.6	78.	78.
		15	0.2	0.4	24.8	77.	77.
		16	0.3	0.5	30.6	82.	82.
		17	0.2	0.4	24.0	79.	79.
		18	0.3	0.4	27.3	95.	95.
		19	0.2	0.5	26.2	92.	92.
		20	0.2	0.4	26.1	77.	77.
		21	0.3	0.5	32.1	90.	90.
		22	0.3	0.5	33.9	82.	82.
		23	0.2	0.4	26.0	78.	78.
		24	0.3	0.5	27.2	98.	97.
		25	0.3	0.5	31.6	84.	83.

3	1	1	1.4	1.2	168.4	84.	83.
		2	4.1	2.2	434.7	94.	91.
		3	3.0	1.4	320.2	94.	93.
		4	2.5	2.0	354.1	71.	70.
		5	4.6	1.7	529.9	87.	85.
		6	1.1	1.5	143.9	74.	72.
		7	3.4	2.0	367.8	93.	88.
		8	2.4	2.1	300.5	80.	80.
		9	0.8	0.8	104.6	79.	79.
		10	3.0	1.6	318.9	95.	92.
		11	4.3	2.7	474.8	91.	91.
		12	1.1	1.0	119.3	92.	91.
		13	1.2	0.9	138.3	85.	82.
		14	1.2	1.2	156.7	79.	75.
		15	1.0	1.2	123.2	81.	81.
		16	1.4	1.4	173.1	82.	80.
		17	2.1	1.3	252.2	82.	79.
		18	5.6	2.9	567.3	99.	95.
		19	2.8	1.9	297.3	95.	91.
		20	1.2	1.3	152.9	80.	79.
		21	3.5	2.4	378.2	92.	86.
		22	3.0	2.0	340.4	88.	87.
		23	1.4	1.2	177.4	80.	78.
		24	2.2	1.6	225.4	97.	91.
		25	1.5	1.7	169.2	90.	88.

Appendix 7

Typical combination report from paired replications

Set 1 PARTC1

Machine states						
1	Mean	45.3	37.9	12.1	4.6	0.0
	St.Dv	0.2	0.2	0.1	0.0	0.0
	Conf.Int	45.8	38.3	12.2	4.7	0.0
		44.8	37.6	12.0	4.6	0.0
2	Mean	30.3	52.6	11.0	6.1	0.0
	St.Dv	0.3	0.2	0.1	0.0	0.0
	Conf.Int	30.9	53.1	11.1	6.1	0.0
		29.7	52.1	10.9	6.0	0.0
3	Mean	69.2	18.6	9.6	2.4	0.0
	St.Dv	0.1	0.1	0.1	0.0	0.0
	Conf.Int	69.5	18.8	9.7	2.5	0.0
		68.9	18.4	9.5	2.4	0.0
4	Mean	30.9	52.8	9.6	6.7	0.0
	St.Dv	0.3	0.2	0.1	0.0	0.0
	Conf.Int	31.5	53.3	9.7	6.7	0.0
		30.2	52.3	9.5	6.6	0.0
5	Mean	44.5	40.8	9.6	5.0	0.0
	St.Dv	0.3	0.2	0.1	0.0	0.0
	Conf.Int	45.1	41.3	9.7	5.1	0.0
		43.9	40.4	9.5	5.0	0.0
6	Mean	46.0	37.4	11.8	4.7	0.0
	St.Dv	0.2	0.2	0.1	0.0	0.0
	Conf.Int	46.6	37.8	11.9	4.8	0.0
		45.5	37.0	11.6	4.7	0.0
Batches per set up						
1	Mean	3.3	2.2	1.3	1.2	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	3.3	2.2	1.3	1.2	0.0
		3.3	2.2	1.3	1.2	0.0
2	Mean	1.5	1.2	1.2	1.1	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	1.5	1.2	1.2	1.1	0.0
		1.5	1.2	1.1	1.1	0.0
3	Mean	1.4	1.1	1.1	1.0	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	1.4	1.2	1.1	1.1	0.0
		1.4	1.1	1.1	1.0	0.0
4	Mean	1.4	1.1	1.1	1.0	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	1.4	1.2	1.1	1.1	0.0
		1.4	1.1	1.1	1.0	0.0
5	Mean	1.4	1.1	1.1	1.0	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	1.4	1.2	1.1	1.1	0.0
		1.4	1.1	1.1	1.0	0.0

6	Mean	3.3	2.2	1.3	1.2	0.0	
	St.Dv	0.0	0.0	0.0	0.0	0.0	
	Conf.Int	3.4	2.2	1.3	1.2	0.0	
		3.3	2.2	1.3	1.2	0.0	

Flowtime

1	Mean	686.0	2178.7	173.0	528.6	875.5	0.0
	St.Dv	5.6	122.6	0.0	5.4	6.5	0.0
	Conf.Int	698.7	2455.9	173.0	540.7	890.1	0.0
		673.3	1901.5	173.0	516.5	860.9	0.0
2	Mean	824.3	2373.8	232.6	635.7	1051.1	0.0
	St.Dv	4.2	87.7	2.6	4.5	4.4	0.0
	Conf.Int	833.7	2572.2	238.5	645.9	1061.0	0.0
		814.9	2175.4	226.7	625.5	1041.2	0.0
3	Mean	874.2	2353.9	314.4	745.2	1029.2	0.0
	St.Dv	3.6	101.7	3.3	4.3	6.3	0.0
	Conf.Int	882.3	2584.0	321.8	754.9	1043.4	0.0
		866.1	2123.8	307.0	735.5	1015.0	0.0
4	Mean	901.4	2451.6	347.0	796.7	1027.2	0.0
	St.Dv	5.4	110.5	4.0	5.6	8.0	0.0
	Conf.Int	913.7	2701.7	356.0	809.5	1045.4	0.0
		889.1	2201.5	338.0	783.9	1009.0	0.0

Flowtime standard deviations

1	Mean	404.7	333.0	401.9	0.0
2	Mean	457.7	375.6	444.2	0.0
3	Mean	455.6	400.5	469.2	0.0
4	Mean	456.5	411.9	475.5	0.0

Flowtime breakdown

1	Mean	52.6	30.8	3.8	0.0	12.7
	St.Dv	0.3	0.3	0.1	0.0	0.1
	Conf.Int	53.4	31.4	4.0	0.0	13.0
		51.9	30.2	3.7	0.0	12.5
2	Mean	47.1	37.6	4.7	0.0	10.6
	St.Dv	0.3	0.2	0.1	0.0	0.1
	Conf.Int	47.7	38.0	5.0	0.0	10.7
		46.4	37.2	4.5	0.0	10.5
3	Mean	46.6	38.8	4.6	0.0	10.0
	St.Dv	0.3	0.1	0.1	0.0	0.0
	Conf.Int	47.2	39.1	4.9	0.0	10.1
		46.0	38.5	4.3	0.0	9.9
4	Mean	49.3	36.6	4.4	0.0	9.7
	St.Dv	0.3	0.2	0.1	0.0	0.1
	Conf.Int	49.9	37.1	4.6	0.0	9.8
		48.7	36.2	4.2	0.0	9.5

Output

1	Mean	1644.9	898.2	746.7	0.0
	St.Dv	8.7	4.9	3.9	0.0
	Conf.Int	1664.7	909.2	755.5	0.0

		1625.1	887.2	737.9	0.0
2	Mean	1084.9	592.3	492.6	0.0
	St.Dv	11.0	6.1	4.9	0.0
	Conf.Int	1109.8	606.1	503.7	0.0
		1060.0	578.5	481.5	0.0
3	Mean	898.5	490.6	407.9	0.0
	St.Dv	8.7	4.7	3.9	0.0
	Conf.Int	918.1	501.3	416.8	0.0
		878.9	479.9	399.0	0.0
4	Mean	646.1	352.6	293.5	0.0
	St.Dv	6.3	3.4	2.9	0.0
	Conf.Int	660.4	360.4	300.0	0.0
		631.8	344.8	287.0	0.0

WIP level

1	Mean	2.3
	St.Dv	0.0
	Conf.Int	2.3
		2.2
2	Mean	1.8
	St.Dv	0.0
	Conf.Int	1.9
		1.7
3	Mean	1.6
	St.Dv	0.0
	Conf.Int	1.6
		1.5
4	Mean	1.2
	St.Dv	0.0
	Conf.Int	1.2
		1.1

Set 2 PARTC3

Machine states

1	Mean	45.3	37.9	12.1	4.6	0.0
	St.Dv	0.2	0.2	0.1	0.0	0.0
	Conf.Int	45.8	38.3	12.2	4.7	0.0
		44.8	37.6	12.0	4.6	0.0
2	Mean	30.7	53.4	9.7	6.1	0.0
	St.Dv	0.2	0.2	0.1	0.0	0.0
	Conf.Int	31.3	53.8	9.8	6.2	0.0
		30.2	53.0	9.6	6.1	0.0
3	Mean	70.7	18.8	7.9	2.5	0.0
	St.Dv	0.1	0.1	0.0	0.0	0.0
	Conf.Int	71.0	19.1	8.0	2.5	0.0
		70.4	18.6	7.8	2.4	0.0
4	Mean	31.0	54.3	7.9	6.8	0.0
	St.Dv	0.3	0.2	0.0	0.0	0.0

Conf.Int	31.6	54.9	8.0	6.8	0.0	
	30.4	53.7	7.8	6.7	0.0	
5 Mean	44.7	42.3	7.9	5.1	0.0	
St.Dv	0.3	0.3	0.0	0.0	0.0	
Conf.Int	45.4	42.9	8.0	5.2	0.0	
	43.9	41.7	7.8	5.1	0.0	
6 Mean	46.0	37.5	11.8	4.7	0.0	
St.Dv	0.2	0.1	0.1	0.0	0.0	
Conf.Int	46.5	37.8	11.9	4.8	0.0	
	45.5	37.2	11.6	4.7	0.0	
Batches per set up						
1 Mean	3.3	2.2	1.3	1.2	0.0	
St.Dv	0.0	0.0	0.0	0.0	0.0	
Conf.Int	3.3	2.2	1.3	1.2	0.0	
	3.3	2.2	1.3	1.2	0.0	
2 Mean	1.7	1.4	1.3	1.2	0.0	
St.Dv	0.0	0.0	0.0	0.0	0.0	
Conf.Int	1.7	1.4	1.3	1.2	0.0	
	1.6	1.3	1.3	1.2	0.0	
3 Mean	1.9	1.4	1.3	1.2	0.0	
St.Dv	0.0	0.0	0.0	0.0	0.0	
Conf.Int	2.0	1.5	1.3	1.2	0.0	
	1.8	1.4	1.2	1.2	0.0	
4 Mean	1.9	1.4	1.3	1.2	0.0	
St.Dv	0.0	0.0	0.0	0.0	0.0	
Conf.Int	2.0	1.5	1.3	1.2	0.0	
	1.8	1.4	1.2	1.2	0.0	
5 Mean	1.9	1.4	1.3	1.2	0.0	
St.Dv	0.0	0.0	0.0	0.0	0.0	
Conf.Int	2.0	1.5	1.3	1.2	0.0	
	1.8	1.4	1.2	1.2	0.0	
6 Mean	3.3	2.2	1.3	1.2	0.0	
St.Dv	0.0	0.0	0.0	0.0	0.0	
Conf.Int	3.4	2.3	1.3	1.2	0.0	
	3.3	2.2	1.3	1.2	0.0	
Flowtime						
1 Mean	678.8	2143.1	173.0	443.8	910.7	0.0
St.Dv	5.3	50.4	0.0	5.4	5.9	0.0
Conf.Int	690.9	2257.1	173.0	456.1	924.2	0.0
	666.7	2029.1	173.0	431.5	897.2	0.0
2 Mean	810.2	2404.7	226.2	563.5	1080.8	0.0
St.Dv	4.5	42.1	2.1	3.8	7.7	0.0
Conf.Int	820.5	2499.9	231.0	572.0	1098.3	0.0
	799.9	2309.5	221.4	555.0	1063.3	0.0
3 Mean	846.1	2408.9	306.0	698.9	1051.4	0.0
St.Dv	6.7	64.6	2.5	5.6	6.8	0.0
Conf.Int	861.2	2555.1	311.6	711.6	1066.9	0.0

		831.0	2262.7	300.4	686.2	1035.9	0.0
4	Mean	859.5	2277.3	332.4	748.7	1013.9	0.0
	St.Dv	5.5	64.5	3.4	5.6	8.6	0.0
	Conf.Int	871.9	2423.3	340.1	761.4	1033.2	0.0
		847.1	2131.3	324.7	736.0	994.6	0.0

Flowtime standard deviations

1	Mean	423.6	277.4	414.7	0.0
2	Mean	471.5	328.2	455.6	0.0
3	Mean	455.1	370.9	481.4	0.0
4	Mean	446.8	386.5	477.6	0.0

Flowtime breakdown

1	Mean	49.9	32.6	4.1	0.0	13.3
	St.Dv	0.3	0.3	0.1	0.0	0.1
	Conf.Int	50.6	33.2	4.3	0.0	13.5
		49.3	32.0	3.9	0.0	13.1
2	Mean	45.2	39.0	4.8	0.0	10.9
	St.Dv	0.4	0.2	0.1	0.0	0.1
	Conf.Int	46.0	39.5	5.1	0.0	11.1
		44.4	38.6	4.5	0.0	10.8
3	Mean	45.6	39.6	4.8	0.0	10.0
	St.Dv	0.4	0.3	0.1	0.0	0.1
	Conf.Int	46.4	40.1	5.0	0.0	10.2
		44.7	39.0	4.6	0.0	9.9
4	Mean	47.2	38.4	4.5	0.0	9.9
	St.Dv	0.3	0.3	0.1	0.0	0.1
	Conf.Int	47.9	39.0	4.8	0.0	10.1
		46.4	37.8	4.3	0.0	9.8

Output

1	Mean	1644.1	815.7	828.4	0.0
	St.Dv	8.4	5.9	8.3	0.0
	Conf.Int	1663.1	829.1	847.2	0.0
		1625.1	802.3	809.6	0.0
2	Mean	1085.6	567.7	517.9	0.0
	St.Dv	11.1	6.7	5.8	0.0
	Conf.Int	1110.7	582.8	530.9	0.0
		1060.5	552.6	504.9	0.0
3	Mean	898.3	523.4	374.9	0.0
	St.Dv	8.8	5.1	7.1	0.0
	Conf.Int	918.1	535.0	390.9	0.0
		878.5	511.8	358.9	0.0
4	Mean	646.3	376.3	270.0	0.0
	St.Dv	6.4	4.5	4.4	0.0
	Conf.Int	660.8	386.4	279.9	0.0
		631.8	366.2	260.1	0.0

WIP level

1	Mean	2.3
	St.Dv	0.0

	Conf.Int	2.3
		2.2
2	Mean	1.8
	St.Dv	0.0
	Conf.Int	1.8
		1.7
3	Mean	1.5
	St.Dv	0.0
	Conf.Int	1.6
		1.5
4	Mean	1.1
	St.Dv	0.0
	Conf.Int	1.2
		1.1

Appendix 8

Typical combination report comparing paired replications

PARTC3 - PARTC1

Machine states

1	Mean	0.0	0.0	0.0	0.0	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0
2	Mean	0.4	0.8	-1.3	0.1	0.0
	St.Dv	0.0	0.1	0.1	0.0	0.0
	Conf.Int	0.4	1.0	-1.2	0.1	0.0
		0.4	0.6	-1.4	0.0	0.0
3	Mean	1.5	0.2	-1.7	0.0	0.0
	St.Dv	0.0	0.0	0.1	0.0	0.0
	Conf.Int	1.5	0.3	-1.6	0.1	0.0
		1.4	0.1	-1.9	0.0	0.0
4	Mean	0.2	1.5	-1.7	0.1	0.0
	St.Dv	0.0	0.1	0.1	0.0	0.0
	Conf.Int	0.2	1.6	-1.6	0.1	0.0
		0.1	1.3	-1.9	0.1	0.0
5	Mean	0.2	1.4	-1.7	0.1	0.0
	St.Dv	0.0	0.1	0.1	0.0	0.0
	Conf.Int	0.2	1.7	-1.6	0.2	0.0
		0.1	1.2	-1.9	0.1	0.0
6	Mean	-0.1	0.1	0.0	0.0	0.0
	St.Dv	0.1	0.0	0.0	0.0	0.0
	Conf.Int	0.1	0.2	0.1	0.0	0.0
		-0.2	0.0	-0.1	0.0	0.0

Batches per set up

1	Mean	0.0	0.0	0.0	0.0	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0
2	Mean	0.2	0.2	0.1	0.1	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	0.2	0.2	0.2	0.1	0.0
		0.1	0.1	0.1	0.1	0.0
3	Mean	0.5	0.3	0.2	0.2	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	0.6	0.4	0.2	0.2	0.0
		0.4	0.3	0.1	0.1	0.0
4	Mean	0.5	0.3	0.2	0.2	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	0.6	0.4	0.2	0.2	0.0
		0.4	0.3	0.1	0.1	0.0
5	Mean	0.5	0.3	0.2	0.2	0.0
	St.Dv	0.0	0.0	0.0	0.0	0.0
	Conf.Int	0.6	0.4	0.2	0.2	0.0
		0.4	0.3	0.1	0.1	0.0

6	Mean	0.0	0.0	0.0	0.0	0.0	
	St.Dv	0.0	0.0	0.0	0.0	0.0	
	Conf.Int	0.0	0.1	0.0	0.0	0.0	
		0.0	0.0	0.0	0.0	0.0	
Flowtime							
1	Mean	-7.2	-35.6	0.0	-84.8	35.2	0.0
	St.Dv	2.9	89.7	0.0	3.6	3.8	0.0
	Conf.Int	-0.7	167.3	0.0	-76.6	43.7	0.0
		-13.7	-238.5	0.0	-93.0	26.7	0.0
2	Mean	-14.1	30.9	-6.4	-72.2	29.7	0.0
	St.Dv	2.5	111.7	4.3	4.5	5.5	0.0
	Conf.Int	-8.4	283.5	3.3	-62.0	42.1	0.0
		-19.8	-221.7	-16.1	-82.4	17.3	0.0
3	Mean	-28.1	55.0	-8.4	-46.3	22.2	0.0
	St.Dv	4.4	132.0	3.0	6.0	7.2	0.0
	Conf.Int	-18.1	353.5	-1.6	-32.7	38.4	0.0
		-38.1	-243.5	-15.2	-59.9	6.0	0.0
4	Mean	-41.9	-174.3	-14.6	-48.0	-13.3	0.0
	St.Dv	3.1	152.2	6.8	5.8	9.7	0.0
	Conf.Int	-34.8	170.0	0.7	-34.9	8.7	0.0
		-49.0	-518.6	-29.9	-61.1	-35.3	0.0
Flowtime standard deviations							
1	Mean	18.9	-55.6	12.8	0.0		
2	Mean	13.8	-47.4	11.4	0.0		
3	Mean	-0.6	-29.7	12.1	0.0		
4	Mean	-9.8	-25.4	2.1	0.0		
Flowtime breakdown							
1	Mean	-2.7	1.8	0.2	0.0	0.6	
	St.Dv	0.2	0.1	0.1	0.0	0.0	
	Conf.Int	-2.3	2.1	0.4	0.0	0.7	
		-3.1	1.6	0.1	0.0	0.5	
2	Mean	-1.9	1.5	0.1	0.0	0.3	
	St.Dv	0.2	0.1	0.1	0.0	0.0	
	Conf.Int	-1.5	1.7	0.3	0.0	0.4	
		-2.2	1.2	-0.2	0.0	0.3	
3	Mean	-1.0	0.7	0.2	0.0	0.1	
	St.Dv	0.2	0.2	0.1	0.0	0.0	
	Conf.Int	-0.5	1.1	0.4	0.0	0.1	
		-1.6	0.4	0.0	0.0	0.0	
4	Mean	-2.1	1.8	0.1	0.0	0.2	
	St.Dv	0.3	0.1	0.1	0.0	0.1	
	Conf.Int	-1.5	2.1	0.4	0.0	0.3	
		-2.7	1.5	-0.1	0.0	0.1	
Output							
1	Mean	-0.8	-82.5	81.7	0.0		
	St.Dv	0.5	6.2	6.0	0.0		
	Conf.Int	0.4	-68.5	95.4	0.0		

		-2.0	-96.5	68.0	0.0
2	Mean	0.7	-24.6	25.3	0.0
	St.Dv	0.4	3.0	2.8	0.0
	Conf.Int	1.6	-17.8	31.6	0.0
		-0.2	-31.4	19.0	0.0
3	Mean	-0.2	32.8	-33.0	0.0
	St.Dv	0.3	4.4	4.6	0.0
	Conf.Int	0.5	42.8	-22.7	0.0
		-0.9	22.8	-43.3	0.0
4	Mean	0.2	23.7	-23.5	0.0
	St.Dv	0.2	2.9	3.0	0.0
	Conf.Int	0.8	30.4	-16.7	0.0
		-0.4	17.0	-30.3	0.0

WIP level		
1	Mean	0.0
	St.Dv	0.0
	Conf.Int	0.0
		0.0
2	Mean	0.0
	St.Dv	0.0
	Conf.Int	0.0
		-0.1
3	Mean	-0.1
	St.Dv	0.0
	Conf.Int	0.0
		-0.1
4	Mean	-0.1
	St.Dv	0.0
	Conf.Int	0.0
		-0.1